UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
ENGINEERING

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.
To renew call Telephone Center, 333-8400

ENGINEERING

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

PHOTO REPRODUCTION
AUG 0 9 REC'D

OCT 1 7 1985
NOV 1 2 REC'D
PHOTO REPRODUCTION

SEP 2 REC'D
JAN 1 0 1989
REC'T DEC 1 7 '88

PHOTO REPRODUCTION
MAR 9 REC'D

L161—O-1096

Engin

26 1979
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

Proceedings

SIXTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

October 4-6, 1978

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

PROCEEDINGS

SIXTEENTH ANNUAL ALLERTON CONFERENCE
ON COMMUNICATION, CONTROL, AND COMPUTING

M. B. Pursley
J. B. Cruz, Jr.
Conference Co-Chairmen

Conference held

October 4-6, 1978
Allerton House
Monticello, Illinois

Sponsored by

The Department of Electrical Engineering
and
The Coordinated Science Laboratory
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

ENGINEERING LIBRARY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

621.3811
Al5
1978

ENGINEERING LIBRARY

FOREWORD

The Sixteenth Annual Allerton Conference on Communication, Control, and Computing was held at the Allerton House on October 4-6, 1978. The Conference was sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the Conference.

In addition to the large number of submitted papers, there were several organized sessions of invited papers. These were five sessions on Computational Complexity organized by F. P. Preparata, and one session each on Communication Theory, organized by L. B. Milstein, Rate-Distortion Theory and Source Coding Techniques, organized by D. L. Neuhoff, Microprocessors in the Control Laboratory, organized by S. Kahne, Singular-Value Analysis for Multivariable Control, organized by A. J. Laub, N. R. Sandell, and G. Stein, Nonlinear Feedback Control, organized by L. Shaw, and Power Systems, organized by P. W. Sauer. We are grateful to these organizers for their time and effort on behalf of the Conference. We are also grateful to the other session chairmen and authors for their contributions to the success of the Allerton Conference.

On Wednesday evening J. L. Doob delivered the plenary address entitled "Modern Developments in Stochastic Processes." This meeting was co-sponsored by the Joint Chapter of the IEEE Information Theory Group and was chaired by its vice-chairman D. L. Cohn.

The support and encouragement of Professor E. C. Jordan, Head of the Department of Electrical Engineering and Professor R. T. Chien, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We thank our colleagues for their generous assistance in organizing the Conference: J. A. Abraham, J. Ackermann, D. P. Bertsekas, E. S. Davidson, A. H. Haddad, W. K. Jenkins, P. V. Kokotovic, D. P. Looze, R. J. McEliece, J. Medanic, G. Metze, D. E. Muller, W. R. Perkins, H. V. Poor, F. P. Preparata, B. R. Rau, D. V. Sarwate, P. W. Sauer, S. Swamy, T. N. Trick, M. E. Van Valkenburg, and N. Wax.

We are grateful to Harold Lawler for his handling of the business aspects of the Conference. Thanks are also due to F. D. Garber and G. Papavassilopoulos for coordinating the projectionists and the visual aid equipment. Special thanks are due to Rose Harris, Rosa Townsend, and Phyllis Young for their expert and skillful coordination of the Conference arrangements. Also we want to thank Rose Harris for her help in the preparation and editing of the Conference PROCEEDINGS.

M. B. Pursley
J. B. Cruz, Jr.

TABLE OF CONTENTS

FOREWORD iii

I-A: MICROPROCESSORS IN THE CONTROL LABORATORY

Chairman and Organizer: S. Kahne
Case Western Reserve University

A MULTI-MICROCOMPUTER FACILITY FOR A DISTRIBUTED COMPUTER CONTROL LABORATORY 1
M. R. Buchner

MICROPROCESSOR LABORATORIES FOR CONTROL 11
M. J. Wozny

A SENIOR-LEVEL COURSE ON MICROPROCESSOR BASED CONTROL SYSTEMS 14
F. A. Briggs and A. J. Koivo

A USER ORIENTED MICROCOMPUTER FACILITY FOR DESIGNING LINEAR QUADRATIC GAUSSIAN FEEDBACK COMPENSATORS 20
P. K. Houpt, J. Wahid, T. L. Johnson,
and S. A. Ward

EFFECT OF FINITE WORD LENGTH IN DISCRETE-TIME CONTROL 30
M. Ahmed and P. R. Belanger

I-B: COMPUTATIONAL COMPLEXITY I

Chairman: E. M. Reingold
University of Illinois

Organizer: F. P. Preparata
University of Illinois

DECOMPOSABLE SEARCHING PROBLEMS 40
J. L. Bentley

A LOWER WORST-CASE COMPLEXITY FOR SEARCHING A DICTIONARY 50
D. S. Hirschberg

INFERRING A TREE FROM LOWEST COMMON ANCESTORS WITH AN APPLICATION TO THE OPTIMIZATION OF RELATIONAL EXPRESSIONS 54
A. V. Aho, Y. Sagiv, T. G. Szymanski,
and J. D. Ullman

SORTING NUMBERS IN LINEAR EXPECTED TIME AND CONSTANT EXTRA SPACE 64
D. B. Johnson and T. F. Gonzalez

HAMILTONIAN PATHS ON A RECTANGULAR CHESSBOARD 73
F. Luccio and C. Mugnai

I-C: DETECTION, ESTIMATION, AND FILTERING

Chairman: S. C. Schwartz
Princeton University

SEQUENTIAL DETECTION WITH MARKOV INTERRUPTED OBSERVATIONS 79
M. T. Hadidi and S. C. Schwartz

OPTIMAL FILTERS FOR NILPOTENT ASSOCIATE-ALGEBRAIC BILINEAR SYSTEMS 89
S. D. Chikte and J. T.-H. Lo

ON MINIMUM LINEAR MEAN-SQUARE ESTIMATION WITH NO MULTIPLICATION 99
T. Y. Yan and K. Yao

SADDLE POINT APPROXIMATION FOR THE ERROR PROBABILITY IN PAM SYSTEMS WITH INTERSYMBOL INTERFERENCE 107
O.-C. Yue

A NOVEL APPROACH FOR DESIGNING NONLINEAR DISCRETE TIME FILTERS: PART I 117
D. Minoo-Hamedani, G. L. Wise, N. C. Gallagher, and T. E. McCannon

A NOVEL APPROACH FOR DESIGNING NONLINEAR DISCRETE TIME FILTERS: PART II 127
T. E. McCannon, N. C. Gallagher, G. L. Wise, and D. Minoo-Hamedani

I-D: FAULT TOLERANT COMPUTING

Chairman: G. Metze
University of Illinois

DECOMPOSITION ALGORITHMS AND SENSITIVITY ANALYSIS FOR PROBABILISTIC CIRCUITS AND FAULT TREES 136
A. S. Rosenthal

A FAULT DETECTING TEST FOR SEMICONDUCTOR RANDOM ACCESS MEMORIES 137
D. S. Suk and S. M. Reddy

EMPIRICAL COMPUTER SYSTEM DIAGNOSIS 138
J. E. Smith

THE TESTABILITY OF TMR NETWORKS 146
J. C. Sutton, III and J. G. Bredeson

A TESTABILITY MEASURE BASED ON INFORMATION THEORY 147
J. A. Dussault

ROVING DIAGNOSIS: DETERMINISTIC ANALYSIS OF A SCHEME FOR SYSTEM SELF-DIAGNOSIS 155
R. Nair, G. Metze, and J. Abraham

ON THE DETECTION OF FAULT-FREE UNITS IN A SYSTEM 156
R. Nair, G. Metze, and J. Abraham

I-E: DYNAMIC SYSTEMS I

Chairman: Nelson Wax
University of Illinois

INVARIANCE AND FEEDBACK FOR FINITE-STATE MACHINES 157
P. E. Liepa and W. M. Wonham

CELLWORK TOPOLOGY, ITS NETWORK DUALS AND SOME APPLICATIONS, THREE DIMENSIONAL KARNAUGH MAP AND ITS VIRTUAL PLANAR REPRESENTATION 165
H. L. Liu and K. S. Fu

PT-MARKED GRAPHS: A REDUCED MODEL OF PETRI NETS 175
S. H. Yu and T. Murata

AUTOMATION PRINCIPLES FOR THE MANAGEMENT OF THE CRITICALLY ILL 185
N. DeClaris and R. A. Cowley

II-A: ESTIMATION AND CONTROL OF STOCHASTIC SYSTEMS

Chairman: A. H. Haddad
University of Illinois

ON THE REPRESENTATION OF OPTIMAL FIXED-LAG SMOOTHERS FOR CONTINUOUS-TIME RANDOM PROCESSES 192
J. S. Meditch

ESTIMATION OF THE ORBITAL STATES OF A SATELLITE FROM GROUND-BASED MEASUREMENTS 202
S. A. Azim and N. K. Sinha

COMPARATIVE EVALUATION OF A UD-FACTORIZED FILTER IN A DIGITAL FLIGHT CONTROL SYSTEM 213
D. Tabak and R. C. Montgomery

II-B: COMPUTATIONAL COMPLEXITY II

Chairman: C. L. Liu
University of Illinois

Organizer: F. P. Preparata
University of Illinois

FLOWSHOP SCHEDULING WITH LIMITED TEMPORARY STORAGE 214
C. H. Papadimitriou and P. C. Kanellakis

OPTIMAL AND NEAR-OPTIMAL SCHEDULING ALGORITHMS FOR BATCHED PROCESSING IN LINEAR STORAGE 224
J. R. Bitner and C. K. Wong

DETERMINISTIC SCHEDULING WITH PIPELINED PROCESSORS 232
J. Bruno, J. Jones, and K. So

FINDING THE BEST SOLUTIONS: THE COST ≤ C PROBLEM AND FINDING MATCHINGS 242
H. N. Gabow

OPTIMAL ALGORITHMS FOR SENSITIVITY ANALYSIS IN ASSOCIATIVE MULTIPLICATION PROBLEMS 252
A. Rosenthal

II-C: MULTIPLE-TERMINAL COMMUNICATIONS

Chairman: W. L. Root
University of Michigan

AN OPTIMAL SCHEME FOR ADDRESSING IN DIGITAL LOOP SYSTEMS 261
J. P. Bhullar and J. B. Anderson

A DECENTRALIZED SHORTEST PATH ALGORITHM 271
J. M. Abram and I. B. Rhodes

RECURSIVE EVALUATION OF OVERALL NETWORK RELIABILITY 278
T. S. El-Sheikh and G. A. Cheston

II-D: INTERACTIVE AND SIMULATION-BASED OPTIMIZATION AND DESIGN

Chairman: M. Wozny
Rennselaer Polytechnic Institute

AN INTERACTIVE CAD PROGRAM FOR CONTROL SYSTEM ANALYSIS AND SYNTHESIS 287
S. J. Larimer and G. B. Lamont

AN EFFICIENT INTERACTIVE OPTIMIZATION METHOD FOR MULTI-OBJECTIVE DESIGN PROBLEMS 296
A. N. Payne and E. Polak

A SIMULATION-BASED METHOD FOR CONTROL SYSTEM PARAMETER OPTIMIZATION 306
D. Crevier and M. A. Nourmoussavi

II-E: DIGITAL TECHNIQUES AND DIGITAL SYSTEMS

Chairman: J. A. Abraham
University of Illinois

A MODEL FOR CONSERVATIVE MAGNETIC BUBBLE LOGIC CIRCUITS 314
W. Coy

SOME PROPERTIES OF THE NBS DATA ENCRYPTION STANDARD 315
J. P. Robinson

MINTERM EXCHANGE IN SPECTRAL LOGIC DESIGN 316
C. Moraga

A UNIQUE INTERFACE TO IMS 324
J. J. Mayewski

A FAST REARRANGEMENT TECHNIQUE ON THE ROW-COLUMN INDEXING SCHEME FOR THE SOLUTION OF SPARSE SYSTEMS OF EQUATIONS 325
E. Nakamae, H. Yamashita, and T. Nomura

III-A: A PROPOSED NATIONAL SYSTEMS CENTER

Chairman: J. S. Meditch
University of Washington

A NATIONAL SYSTEMS CENTER PROPOSAL 334
S. Kahne

III-C: COMMUNICATIONS AND CODING

Chairman: D. L. Cohn
University of Notre Dame

QUANTIZATION LOSS IN OPTICAL COMMUNICATION SYSTEMS 344
D. L. Snyder and I. B. Rhodes

ASSESSMENT OF CODING POWER GAIN ON A SATELLITE COMMUNICATIONS LINK 345
P. K. Lee

VITERBI DECODER SYNCHRONIZATION AND BER MEASUREMENT USING METRIC VALUES 354
R. W. Koralek and M. A. Metzler

CONTROL THEORY APPLIED TO THE PROBLEM OF REDUCING INTERMODULATION DISTORTION IN NONLINEAR AMPLIFIERS UNDER MULTICARRIER OPERATION 364
M. Blanke, J. Langeland-Knudsen, and T. P. McElhone

III-D: DIGITAL FILTERS

Chairman: W. K. Jenkins
University of Illinois

LOW-NOISE REALIZATIONS FOR DIGITAL FILTERS WITH POLES NEAR THE UNIT CIRCLE 372
D. C. Munson, Jr. and B. Liu

A SET OF LINEAR TRANSFORMATIONS FOR HIGH MODULARITY DIGITAL AND ACTIVE FILTERS 382
A. M. Ali

CONSTRAINED LINEAR PREDICTION: PRELIMINARY RESULTS 391
N. R. Malik

IV-A: COMMUNICATION SYSTEMS

Chairman and Organizer: L. B. Milstein
University of California, San Diego

COMBINED SOURCE-CHANNEL CODING OF IMAGES 392
J. W. Modestino and D. G. Daut

ASYMPTOTIC DISTRIBUTION OF THE SAMPLE COHERENCE USING OVERLAP PROCESSING 403
R. Lugannani

EVALUATION OF ERROR PROBABILITIES FOR MPSK SATELLITE COMMUNICATION SYSTEMS WITH INTERFERENCE 413
J. K. Omura, T. C. Huang, and P. C. Jain

PERFORMANCE OF PSK SYSTEMS WITH BANDPASS LIMITERS 423
K. Yao and L. B. Milstein

IV-B: SINGULAR-VALUE ANALYSIS FOR MULTIVARIABLE CONTROL

Co-Chairmen: A. J. Laub and N. R. Sandell, Jr.
Massachusetts Institute of Technology

Co-Organizers: A. J. Laub and N. R. Sandell, Jr.
Massachusetts Institute of Technology
G. Stein
Honeywell Systems Research Center

COMPUTATIONAL ASPECTS OF THE SINGULAR VALUE DECOMPOSITION AND SOME APPLICATIONS 432
A. J. Laub

COMPUTATIONAL PROBLEMS WITH MODAL ANALYSIS 443
B. C. Moore

TIGHT BOUNDS ON THE RESPONSE OF MULTIVARIABLE SYSTEMS WITH COMPONENT UNCERTAINTY 451
M. G. Safonov

SINGULAR VALUES AND FEEDBACK: DESIGN EXAMPLES 461
G. Stein and J. C. Doyle

ROBUST STABILITY OF MULTIVARIABLE FEEDBACK SYSTEMS 471
N. R. Sandell, Jr.

IV-C: FOUNDATIONS OF COMPUTING

Chairman: D. E. Muller
University of Illinois

Organizer: F. P. Preparata
University of Illinois

SUCCINCTNESS OF DEGREE AUTOMATA AND PROBABILISTIC AUTOMATA 480
C. M. R. Kintala and D. Wotschke

ON LANGUAGES ACCEPTED BY TIME-BOUNDED ORACLE MACHINES 489
R. V. Book and C. Wrathall

REAL-TIME SIMULATION OF MULTIHEAD RESET TAPE UNITS 495
S. R. Kosaraju

INTERMEDIATE COMPLEXITY IN DATA MATRIX PROBLEMS 504
F. N. Springsteel and P. P. Pudlak

SATISFIABILITY PROBLEMS FOR PROPOSITIONAL CALCULI 513
H. R. Lewis

IV-D: DYNAMIC SYSTEMS II

Chairman: D. P. Looze
University of Illinois

LINEARIZING EFFECT OF FEEDBACK ON NONLINEAR DYNAMICAL SYSTEMS 521
C. A. Desoer, Y. T. Wang, and A. N. Payne

A NECESSARY AND SUFFICIENT CONDITION FOR LOCAL CONSTRAINED CONTROLLABILITY OF A LINEAR SYSTEM 529
B. R. Barmish and W. E. Schmitendorf

STATE CONTROLLABILITY AND OBSERVABILITY FOR LINEAR NEUTRAL SYSTEMS 539
D. A. O'Connor and T. J. Tarn

ON THE INDUCED SUBFLOW IN REGULATOR SYSTEMS ON DIFFERENTIABLE MANIFOLDS 547
J. S. A. Hepburn and W. M. Wonham

ON STABILIZATION OF CONSERVATIVE DYNAMICAL SYSTEMS 557
R. K. Miller and A. N. Michel

ON A NEW CLASS OF SEMI-LINEAR SYSTEMS AND THEIR APPLICATIONS 566
H. Sira-Ramirez

CARDIAD DESIGN: PROGRESS IN THE FOUR INPUT/OUTPUT CASE 567
R. M. Schafer and M. K. Sain

IV-E: PARALLEL PROCESSING

Chairman: B. R. Rau
University of Illinois

A MODEL FOR THE PARALLEL EXECUTION OF PROGRAMS WITH ASYNCHRONOUS OPERATIONS 568
V. Vairavan and K. Vairavan

STING-UNLESS: A CONDITIONAL, INTERLOCK-FREE STORE INSTRUCTION 578
D. P. Friedman and D. S. Wise

PARALLEL SORTING 585
L. D. Coraor and J. P. Robinson

PARTITIONABLE SIMD COMPUTER SYSTEM INTERCONNECTION NETWORK UNIVERSALITY 586
H. J. Siegel

A COMPUTER ARCHITECTURE FOR PARALLEL PROCESSING 596
T. N. Mudge

V-A: COMPUTATIONAL COMPLEXITY II

Chairman: R. J. McEliece
University of Illinois

Organizer: F. P. Preparata
University of Illinois

NEW UPPER BOUNDS ON THE NON-ASYMPTOTIC COMPLEXITY OF MATRIX MULTIPLICATION 597
R. L. Probert

COMPUTING EXTREMAL AND APPROXIMATE DISTANCES IN A GRAPH HAVING UNIT COST EDGES 607
K. S. Booth and R. J. Lipton

A GENERALIZED ALGORITHM FOR CENTRALITY PROBLEMS ON TREES 616
A. Rosenthal

ORTHOGONAL PACKINGS IN TWO DIMENSIONS 626
B. S. Baker, E. G. Coffman, Jr., and R. L. Rivest

A FAMILY OF BOOLEAN FUNCTIONS WITH NONLINEAR FORMULA SIZE 636
Y. Breitbart

V-B: NONLINEAR FEEDBACK CONTROL

Chairman and Organizer: L. Shaw
Polytechnic Institute of New York

A SOLUTION FOR BILINEAR DISCRETE-TIME CONTROL SYSTEMS 645
K. N. Swamy and T. J. Tarn

APPLICATION OF MULTILINEAR ALGEBRA IN OPTIMAL REGULATION OF NONLINEAR POLYNOMIAL SYSTEMS 655
M. R. Buric

LOCAL DECOMPOSABILITY AND THE DISTURBANCE DECOUPLING PROBLEM IN NONLINEAR AUTONOMOUS SYSTEMS 664
S. H. Mikhail and W. M. Wonham

A DOUBLE INTEGRAL QUADRATIC COST AND TOLERANCE OF FEEDBACK NONLINEARITIES 670
W. H. Kwon and A. E. Pearson

A NEWTON-LYAPUNOV DESIGN FOR A CLASS OF NONLINEAR REGULATOR PROBLEMS 679
J. H. Chow

A NONLINEAR SERVO FOR LINEAR SYSTEMS 689
L. Shaw

V-C: RATE-DISTORTION THEORY AND SOURCE CODING TECHNIQUES

Chairman and Organizer: D. L. Neuhoff
University of Michigan

MISMATCH THEOREMS FOR TRELLIS ENCODING COMMUNICATION SYSTEMS 696
J. G. Dunham

ENTROPY-CODED GROUP CODES FOR DATA COMPRESSION 697
A. Leon-Garcia and L. D. Davisson

RATE DISTORTION FUNCTIONS FOR SQUARED ERROR DISTORTION MEASURES 704
S. L. Fix

A NEW RATE-DISTORTION FUNCTION AND NEW CODING RESULTS FOR ANALOG SOURCE DIGITIZATION 712
W. A. Finamore and W. A. Pearlman

A COST FUNCTION FOR SEQUENTIAL CODING 725
J. B. Anderson and S. Mohan

CAUSAL SOURCE CODING 735
D. L. Neuhoff

V-D: CONTROL SYNTHESIS AND DESIGN

Chairman: J. E. Ackermann
University of Illinois and
DFVLR Institute for Dynamics of
Flight Systems, West Germany

ON ROBUST EIGENVALUE/EIGENVECTOR ASSIGNMENT 744
S. Pohjolainen and H. Koivo

A NEW SPECTRAL SYNTHESIS PROCEDURE FOR MULTIVARIABLE REGULATORS 754
R. A. Maynard, R. R. Mielke, S. R. Liberty, and S. Srinathkumar

DECENTRALIZED STABILIZATION OF LARGE-SCALE SYSTEMS USING MODAL CONTROL AND EIGENVALUE SENSITIVITIES 762
Z. Elrazaz and N. K. Sinha

MINIMUM NORM OUTPUT-FEEDBACK POLE ASSIGNMENT 770
G. Klein and B. C. Moore

DESIGN OF REDUCED-ORDER CONTROLLERS FOR LARGE FLEXIBLE SPACE STRUCTURES 780
S. M. Joshi and N. J. Groom

ON A SELF-ADJUSTING CONTROL SYSTEM 792
N. K. Loh and D. H. Chyung

AN ANALYSIS OF IMAGE CONTROL 793
R. D. Bonnell, M. H. Huhns, and S. Renjen

THE SADDLE SET CONCEPT AND ITS APPLICATION IN ADAPTIVE OPTIMAL CONTROL 802
E. Fogel

HIGH ORDER STIFFLY STABLE METHODS FOR ANALYSIS OF NONLINEAR CIRCUITS AND SYSTEMS 814
A. S. C. Sinha and C. F. Yokomoto

VI-A: COMPUTATIONAL COMPLEXITY IV

Chairman: A. C. Yao
Stanford University

Organizer: F. P. Preparata
University of Illinois

A FAST, SPACE-EFFICIENT AVERAGE-CASE ALGORITHM FOR THE 'GREEDY' TRIANGULATION OF A POINT SET, AND A PROOF THAT THE GREEDY TRIANGULATION IS NOT APPROXIMATELY OPTIMAL 824
G. K. Manacher and A. L. Zobrist

GENERALIZED VORONOI DIAGRAM IN THE PLANE 833
R. L. Drysdale, III and D. T. Lee

OPTIMAL EXPECTED-TIME ALGORITHMS FOR CLOSEST-POINT PROBLEMS 843
J. L. Bentley, B. W. Weide, and A. C. Yao

VI-B: POWER SYSTEMS

Chairman: M. S. Chen
University of Texas, Arlington

Organizer: P. W. Sauer
University of Illinois

PARALLEL SOLUTION OF NONLINEAR INITIAL AND MULTIPLE POINT BOUNDARY VALUE PROBLEMS 852
F. L. Alvarado

A NEW APPROACH TO POWER SYSTEM RELIABILITY CALCULATIONS 862
G. T. Heydt and J. B. Franklin

CONTROL CONCEPTS FOR BULK INTERCONNECTED POWER SYSTEMS IN AN EMERGENCY STATE 870
J. Meisel

CONTROL OF THE LARGE INTERCONNECTED POWER SYSTEM IN EMERGENCIES 880
J. Zaborszky

AREA LOAD-FREQUENCY CONTROL 882
T. M. Athay, H. G. Kwatny, and R. G. Smith

PLANNING POWER TRANSMISSION NETWORKS UNDER UNCERTAINTY 894
R. Fischl

VI-C: MODELING AND IDENTIFICATION

Chairman: N. K. Sinha
McMaster University

MODEL REDUCTION OF DYNAMIC SYSTEMS OVER A FREQUENCY INTERVAL 903
G. Langholz and Y. Bistritz

SELECTION OF MODEL STRUCTURE FOR CONTROL 913
P. R. Belanger

A FREQUENCY DESIGN TECHNIQUE FOR MODEL MATCHING AND SENSITIVITY REDUCTION 914
R. M. DeSantis and S. Lefebvre

MODELS FOR THE IDENTIFICATION OF LINEAR MULTIVARIABLE DISCRETE-TIME SYSTEMS 923
H. El-Sherief and N. K. Sinha

AN APPLICATION OF SALUT APPROACH TO THE IDENTIFICATION OF A TWO AREA POWER SYSTEM 933
S. Lefebvre, R. M. DeSantis, G. Salut, and J. Aguilar-Martin

A STEP-BY-STEP PROCEDURE FOR COMPLETE IDENTIFICATION OF CERTAIN MULTIPLE-INPUT/MULTIPLE-OUTPUT STOCHASTIC SYSTEMS 941
E. D. Eyman and E. A. Adedotun

VI-D: SOURCE ENCODING, SPEECH AND SIGNAL PROCESSING

Chairman: J. G. Dunham
Washington University

THE EFFECTS OF RANDOM CHANNEL ERRORS ON ADAPTIVE DELTA MODULATION ENCODED SIGNALS 951
D. P. Prezas and J. L. LoCicero

BRANCHING PROCESS METHODS FOR THE SINGLE STACK ENCODING ALGORITHM 961
S. Mohan and J. B. Anderson

VOICED/UNVOICED/MIXED EXCITATION CLASSIFICATION IN SPEECH ANALYSIS 971
L. J. Siegel

AN ADAPTIVE NOISELESS VARIABLE-TO-FIXED ENCODING ALGORITHM FOR DISCRETE ERGODIC SOURCES 972
L. Th. N. M. Mulder and D. L. Cohn

QUANTIZATION OF BIVARIATE CIRCULARLY SYMMETRIC DENSITIES 982
J. A. Bucklew and N. C. Gallagher

ON KALMAN ESTIMATOR IMPLEMENTATION WITH CTD'S IN TRANSMISSION SYSTEMS 991
L. Bosotti and G. Pirani

"A MULTI-MICROCOMPUTER FACILITY FOR A DISTRIBUTED COMPUTER CONTROL LABORATORY"

MARCUS R. BUCHNER
Systems Engineering, Computer Engineering and Information Sciences
Case Western Reserve University
Cleveland, Ohio 44106

Abstract

The increased interest in and implementation of distributed computer control systems in recent years has prompted considerable quantitative and analytical research in the areas of hierarchical and decentralized control. The multicomputer control/process facility described in this paper provides a convenient testbed for the validation of analytical models and theoretical concepts as well as provide motivation for numerous research ideas. Consisting of a sophisticated two-level multicomputer control system, a microprocessor based serial computer communication subsystem, a discrete event process, and a process/control system interface, the complexity and generality of the facility allows its use in a variety of roles for research involving the analysis and design of distributed computer control systems.

Introduction

With the increased use of computers, particularly microcomputers, for the control of complex technological systems, distributed computer control (DCC), the implementation of control algorithms on a multiple computer system has become a subject of considerable interest.[1,2,3,4] These complex systems have four general properties which serve to characterize a large class of control structures that is important in engineering applications. First, the controller functions are decentralized, i.e., the control calculations, information processing, data acquisition, and control implementation tasks are shared among several devices capable of autonomous operation. Second, distributed computer control systems tend to be dispersed geographically, implying the need for a computer-to-computer data communication subsystem. Third, the algorithms used for the control of complex systems are typically structured in a hierarchical fashion. Thus, lower level functions are characterized by relatively short time frames, simple decision-making functions, and the processing of large quantities of "raw" data, and higher levels of the control hierarchy are characterized by longer time frames, complex decision-making functions and aggregate measures of plant performance. A fourth important property of DCC systems is that they exhibit a potentially wide variety of responses to the occurrence of disturbances entering the system. This includes both small-scale disturbances causing only local effects on system operation, and large-scale disturbances tending to cause structural changes in the system, e.g., processor failure.

As a result of these properties there are many potential advantages of DCC systems such as:[5,6]

(1) Increased fault tolerance, reliability, and availability
(2) Natural implementation of multi-level control algorithms
(3) Modularity, i.e., ease of expansion
(4) Flexibility, i.e., ability to reconfigure hardware and software to meet changing needs

(5) Simplicity in operation, debugging, documentation, and construction
(6) Enhanced safety
(7) Reduced cabling costs, and
(8) Reduced computer system costs.

Many of these characteristics have been demonstrated in isolated application studies and several quantitative studies with grossly simplified model assumptions. At present, the complexity and stochastic nature of these control systems precludes a complete analytical treatment of the effects on global system (the controlled system) performance of many significant attributes of the control system such as communication protocols, processor overhead, computer reliability, hardware/software reconfigurability, and sensor/actuator performance. In modern distributed computer control systems such attributes can have a substantial impact on the expected system performance in the presence of processor/control system disturbances.[7] The approaches taken thus far, however, in the study of microprocessor-based distributed computer control systems have been from three major perspectives: (1) multilevel hierarchical control, (2) decentralized control, and (3) application studies. Although this body of theory, concepts, and applications provides a general framework for describing complex control problems for large-scale systems, it has not yet provided the deep insight necessary for the analysis and design of detailed implementation structures. In response to these observations, a general purpose laboratory facility has been designed and is being implemented to investigate a wide range of behavior of DCC systems. This facility, described in detail in the remainder of the paper, consists of a multicomputer control system interfaced to a discrete event based system to be controlled.

Figure 1 illustrates the global structure of and the interactions among the control system and the controlled process. As can be seen, there are five major components in the facility: (1) a PDP 11/34 minicomputer for executing higher level control functions, (2) three MOS 6502 based microcomputers for low level control and data acquisition tasks, (3) a computer-computer (microprocessor based) serial data highway, (4) the control system/process interface, and (5) the process (a complex HO gauge model railroad).

In this configuration the minicomputers main function is to coordinate the actions of the lower level microcomputer controllers. Typical tasks performed at the upper level include scheduling, route scheduling, path optimization, and dynamic control task reallocation, while at the lower level primary tasks include sensor/actuator control, signal conditioning of raw data, and direct control.[8,9]

The design of the facility has provided a rich experience for the students involved in the project and will continue to provide a flexible "real world" testbed for the validation of theoretical concepts. It is expected that the complete system will be used to model the application of distributed computer control in such diverse areas as manufacturing, transportation, communication, industrial process control, and mining operations.

The Process

In order to demonstrate or investigate general aspects of a distributed computer control system, one should be able to control a wide variety of types of processes. This entails the design of either an extremely flexible control system/process interface or dedicated interface equipment for specific processes. However, because the main use of the facility will be to examine distributed control of complex systems and because there has recently been an increased use of discrete event models to represent complex systems, a process (or system) was needed which could effectively model a large number of types and behavioral characteristics of discrete event systems. By a discrete event system, one means a system in which the state variables change value at discrete instants of time in response to either the occurrence of exogenous events or the scheduling of endogenous events. Thus, discrete event models can provide a suitable framework for representing systems in which significant changes, those affecting future behavior, can occur in response to various internal or external decisions, disturbances, or inputs acting on the system at discrete instants in time.

In particular, with the increased use of hierarchical modelling and control principles, there has been a parallel interest in using discrete event models to represent aggregate behavior of large systems and decision making processes. Example systems which can usefully be modelled in this manner include electric power generation and distribution systems, coal haulage and minimg systems, conveyer belt operations, crane operations, queuing systems, computers, communications systems, and transportation systems.

To obtain a representative discrete event system which is sufficiently complex to admit a broad spectrum of interesting control problems, which is relatively easy to modify and expand, and which provides a visual feedback of system performance, an HO gauge model railroad system was chosen for the process to be controlled. For this system figure 2 depicts the current track layout which contains 80 feet of track and 25 switches. In addition the track is partitioned into three local regions (corresponding to local control regions for the microcomputers) each containing block sections (implementing velocity actuation), sensors (for determining train location and identification information), and which can currently support up to and including form model trains.

This particular layout was chosen for several reasons, among which are:

(1) The layout may be used for both "round and round" (2 basic routes) and "point to point" (3 basic routes) operation.
(2) There are many variations for routes (by using the passing sidings) so that complex scheduling issues are present.
(3) There are about 25 switches so the control is adequately complex.
(4) There is room for the simultaneous control of at least four Locomotives.
(5) There are four "spurs" which will be useful for contingency operations.
(6) There is one "crossing" (90°) which enhances the scheduling complexity.
(7) There are two reversing loops so the trains may go in any direction on any track given any initial configuration. Also block control of the insulated reversing loops is required.

It should be noted that a major effort was required by the project personnel in order to treat the model railroad as being representative of a broad class of discrete event systems and not as being a particluar transportation system. Great care was taken to insure maximum flexibility and generality in the control system so as not to be specific to a train transportation system. The only exception occurred in the process/control system interface which was unavoidably somewhat peculiar to the model railroad system.

With this system there are many interesting control problems which may be addressed. These range from the routing of a single train on a particular path to the coordination of several trains so as to achieve maximum throughput of components. A sampling of additional problems which can be studied include: complex scheduling operations, route optimization, and adaptive behavior, With proper associations, the use of the train system to study these types of problems, in the above described discrete event systems, is immediate.

Communication Subsystem

The addition of a computer to computer communication network to a system of independent computers allows for the realization of the many advantages of distributed computer control systems described earlier.[10] This occurs as a result of the distributed computation, data storage and software/hardware reconfiguration potential of an interconnected system of computers. Clearly, the use of a high speed communication system distinguishes a distributed control system from more traditional independent computer control systems by allowing a global and local coordination of control, decision making, data acquisition, man-machine interface activities, and data base management.

The communication system is responsible for data, program, command, and status information transfer among the computers in the control system. Thus, the main task of the system is to provide for error free transmission and reception of information originating at any device and directed at any other device such that the information retains its original meaning. In order to accomplish this, a general purpose microprocessor based communication interface, a serial data highway, and a microprocessor based highway controller has been designed (see Figure 1).

The functions of the highway/control computer interface are to implement a communication protocol and to free the control computers from communication related tasks. A microprocessor based design was selected so as to allow maximum flexibility in investigating the effects of various protocol designs. As used here, a protocol is a set of rules for operating a communication system to solve operating problems such as framing, error control, sequence control, data transparency, line control, timeout control, and startup control.

The second component in the subsystem, the data highway, will consist of a differential serial data line and a differential clock line interconnecting all interface microprocessors, and two request lines and a grant line connecting the interface microprocessors to the highway controller.

The third major component is the highway controller. As a result of the required flexibility of the control system, this component is also microprocessor based. This allows for different communication network configurations to be easily programmed (dynamically). The controller accepts and arbitrates requests from a prioritized interrupt structure for the use of the data highway. Additionally, the controller processor is responsible for storing and updating statistics of network utilization, e.g. bus utilization, mean queue length, assessing the status of devices attached to the highway, and adjusting the data baud rate.

In short the communications subsystem, a very complex component of the overall distributed computer control system, having been designed with maximum flexibility and reliability, provides for the investigation of the effects of a variety of communication parameters, structures, and protocols on global control system performance.

Process/Control Interface, Sensors and Activators

Controlling the state of a system subject to the occurrence of disturbances requires that information related to the system state be made available to the control system in a timely fashion. To accomplish this in the discrete event model train system, a sensor/detector subsystem was designed to provide train location and identification information to be obtained from sensors located along the track. The information from the sensor is then digitally encoded and temporarily stored in the detector. Next, an interrupt is generated to the control microcomputer, whose interrupt service routine: 1) analyzes the interrupt source (detector subsystem or communication subsystem), (2) examines the data to ascertain if it indicates state information or a faulty sensor, (3) stores the data in an appropriate stack for further processing, and (4) returns control to the microcomputer main program. Further, the detector interface allows the microcomputer to interact with the control detector in the event of the discovery of a fauly sensor or the generation of new state information. The complete philosophy of operation is thus consistent with the idea of distributing the control and data acquisition tasks in the system and frees the microcomputer to execute more appropriate (higher level) tasks.

In order to implement this system, train mounted oscillators (having a unique frequency allocation) radiate signals to track mounted receiving coils (sensors). These sensor coils are directly connected to the detector which scans each of the sensors. In the event that a signal is detected on a sensor, the detector logic locks onto the active sensor, measures the period between two adjacent sensor pulses, and generates an interrupt to the microcomputer.

The sensors (see Figure 3) are scanned using a multiplexor tied to a 4 bit counter that is stopped every time a signal is detected. The first incoming pulse (which halts the scanning process) enables an 8 bit counter which then stops on the occurrence of a second pulse from the multiplexor, an interrupt then being generated for the microcomputer. The microcomputer proceeds to read the status of the two counters and resets the detector allowing the scan to continue at the next sensor. In this fashion the two counters provide location and identification information.

In addition to these capabilities, the microprocessor can command the detector to rescan a particular sensor in the event that a switch reading is suspect, and the detector can locate faulty or noise prone sensors on the track.

As important as the sensor subsystem is for obtaining process state information, the actuator subsystem provides the only means by which the control algorithms, man-machine interface, the communication subsystem, and the state information can affect the process. Both velocity control and switch control components are included in this important component. While velocity control implements the routing and the amount of power delivered to physical track sections as directed from the control microcomputers, the switch control maintains and alters the status of the track switches in the process. As a result of the relatively low power levels, the quantity of information required to control the switches, and the availability of commercially made remote switches, the design and construction of the switch control was straightforward. The microcomputer/switch interface consists primarily of a standard single transistor based switch circuit. On the other hand, the velocity control subsystem design is relatively complex resulting from the high power levels needed to run the model trains, the number and polarity of desired power levels, and the need for interfacing to peripheral I/O parts of the control computers.

The first consideration in designing the power distribution system for the velocity control is the selection of either block or non-block control. Due to the complexity, expense, and sensitivity to noise of the non-block methods, a block control technique was adopted. There are three basic methods of providing block control: relays, high current switching transistors, and logical level switching and power boosters. The first two methods were rejected due to high cost and circuit complexity.

The block control implementation uses a pulse width modulation unit for each engine, a switching network, and a power booster for each block. Although the power booster is duplicated at each block, cost is reduced significantly through the use of a simple implementation requiring only four inexpensive transistors and a few passive components. Each power booster can supply from 0 to 12 volts with a maximum current of 1.2 amps.

The block control implementation uses the following protocol: (See Figure 4 .)

1) The appropriate pulse modulation unit is informed through a speed and direction command word of the velocity to be imparted to the engine.

2) The power supply is also informed through the two block command words to which blocks power should be routed. There are two block command words for each pulse modulation unit and its associated network.

Control Computer Description

As previously noted, there are four control oriented computers in the facility, one minicomputer and three microcomputers. The minicomputer is a PDP 11/34 which contains 64K words of main memory, a floating point processor, and a combination of fixed and removable disk cartridges yielding a bulk storage capability of 7.5 M bytes. In addition there is a line printer, a VT 55 graphics terminal, three LA36 terminals, a laboratory peripheral subsystem, and a serial input/output interface for experimentation and system development. One of the serial interfaces to the DEC Unibus is devoted to a communication subsystem microprocessor interface. Currently, a sophisticated, hierarchical, man machine interface for the 11/34 is being developed.

The microcomputers used for lower level control computation, data acquisition, and signal processing are the Ohio Scientific 504A's which are 6502 based single board computers having 16 lines of programmable digital I/O. There currently is 2K of RAM and 2K of ROM on the boards for system monitor and interface control tasks. Shortly an additional 2K of RAM will be added for the inclusion of various real-line control tasks and a minimal real time operating system.

Future plans for expanding the control system, to accommodate expansion plans for the process, include the addition of microprocessor control computers having direct memory access (DMA) from both the process interface and the communication network interface.

Summary

A complex control system/process has been described which can provide for the validation of distributed computer control concepts and techniques. Additionally, many interesting and important problems, arising from the project, have stimulated research activity. Another paper, in preparation at this time, will discuss many details of the distributed control aspects of the project. While the project is currently ending the design stage and progressing through the implementation stage, the complex nature of the system, the rapidly changing state of the art in microprocessors and data communication, and the desire to study increasingly sophisticated control systems, allows one to consider the project as only beginning.

Acknowledgements

The author would like to acknowledge and thank the many undergraduate students, graduate students, and faculty who laid the foundation for this project. In particular, Professor Irving Lefkowitz who proposed the original idea for the project, provided invaluable guidance and suggestions; Bruce Evans, Rick Dulik, Dave Kovacic, and Don Pratt who were involved in the project from its inception and contributed in many aspects of the design and implementation; and finally those students and faculty who will be involved in ensuing phases of the project, all deserve special recognition.

References

1. Auslander, Takahashi, Tomizuka, "Direct Digital Process Control: Practice and Algorithms for Microprocessor Application", Proceedings of the IEEE, Vol. 66, No. 2, Feb. 1978.

2. Smith, R., "Put Your Control Where the Action Is", Instrum. Contr. Syst., Vol. 49, No. 4, p. 35, April 1975.

3. Wiatrowski, C. and Teeple, C., "Add Flexibility to your Control System with Distributed Data Processing", Instrum. Contr. Syst., Vol. 49, No. 3, p. 37, March 1976.

4. Bailey, S.J., "Direct Digital Control: The On-line Scene Today", Contr. Eng., Vol. 24, No. 1, p. 20, Jan. 1977.

5. Fraade, P.J., et al. "A Survey of Computer Networks and Distributed Control", Digital Computer Applications to Process Control,

IFAC, 1977, North Holland.

6. Steusloff H., "Hardware Structures and Software for Distributed Computer Control Systems", Purdue Conference on Hierarchical Computer Control, 1976.

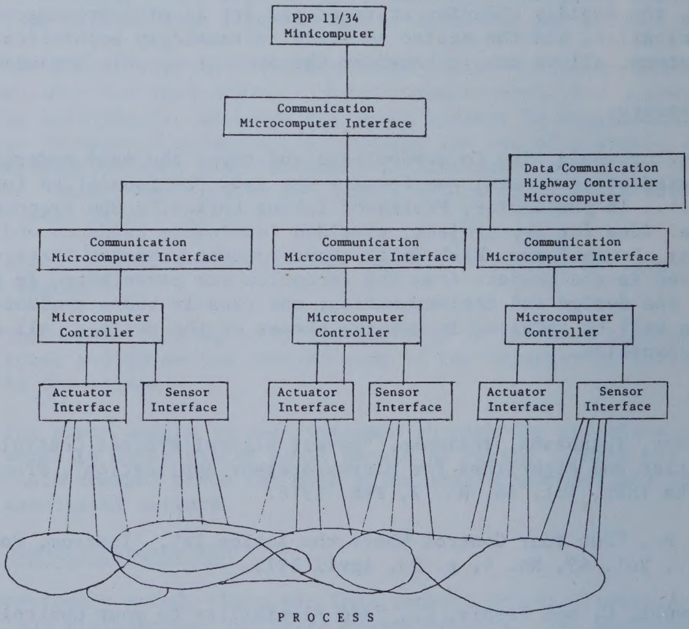
7. Engelberg, Howard, Mellichamp, "A Real-time Simulation of a Hierarchical Computer Network for Process Control", pp. 37-49, Simulation, Feb. 1978.

8. Mesarovic, Macko, Takahara, Theory of Hierarchical Multilevel Systems, Academic Press, 1970.

9. Lefkowitz, I., "Integrated Control of Industrial Systems", Phil. Trans. R. Soc. London, A 287, pp. 443-465, 1977.

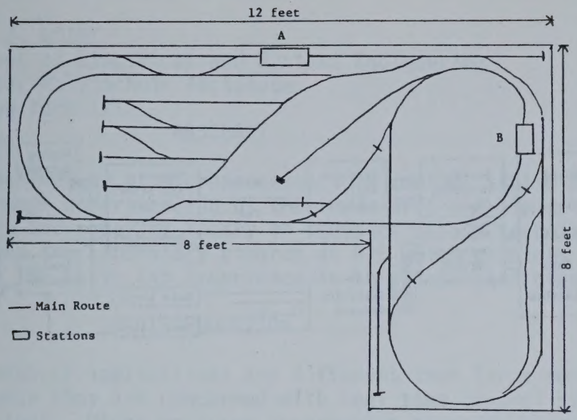
10. Rose, C.W., Schoeffler, J., "Distribution of Intelligence and Input/Output in Data Acquisition Systems", Proc. International Telemetering Conference, Vol. XII, pp. 705-720.

Figures


PDP 11/34
Minicomputer
Communication
Microcomputer Interface
Data Communication
Highway Controller
Microcomputer
Communication
Microcomputer Interface
Communication
Microcomputer Interface
Communication
Microcomputer Interface
Microcomputer
Controller
Microcomputer
Controller
Microcomputer
Controller
Actuator
Interface
Sensor
Interface
Actuator
Interface
Sensor
Interface
Actuator
Interface
Sensor
Interface
PROCESS

Distributed Computer Control/Process Block Diagram

Figure 1


12 feet
A
B
8 feet
8 feet
Main Route
Stations

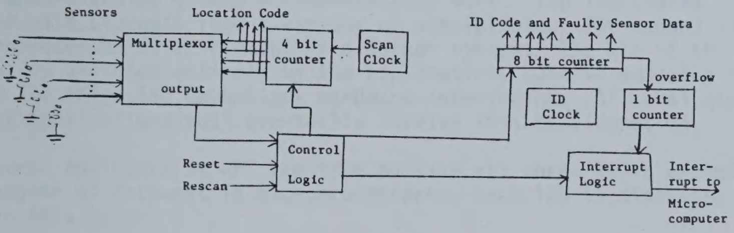
HO Gauge Model Train Track Layout

Figure 2

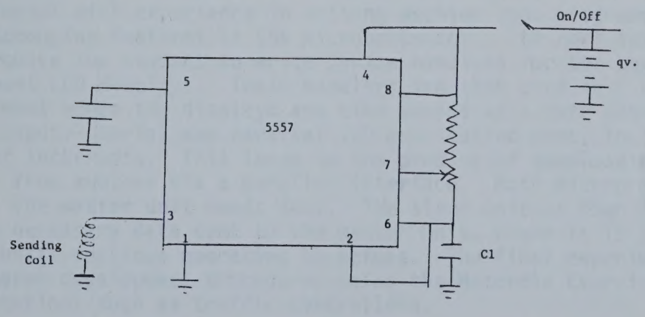
Figure 3

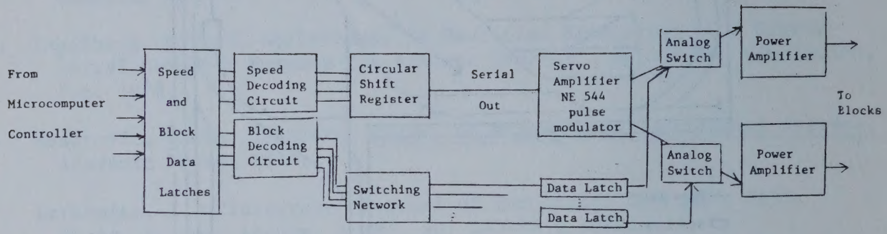
Sensor/Detector/Transmitter Block Diagram

Detector/Sensor


Sensors
Location Code
Multiplexor
output
4 bit counter
Scan Clock
ID Code and Faulty Sensor Data
8 bit counter
overflow
ID Clock
1 bit counter
Control Logic
Reset
Rescan
Interrupt Logic
Inter-rupt to
Micro-computer

Transmitter


On/Off
qv.
4
5
8
5557
7
3
6
1
2
Sending Coil
C1


From
Microcomputer
Controller
Speed
and
Block
Data
Latches
Speed
Decoding
Circuit
Block
Decoding
Circuit
Circular
Shift
Register
Serial
Out
Switching
Network
Servo
Amplifier
NE 544
pulse
modulator
Data Latch
Data Latch
Analog
Switch
Analog
Switch
Power
Amplifier
Power
Amplifier
To
Blocks

Block Control Implementation

Figure 4

MICROPROCESSOR LABORATORIES FOR CONTROL

MICHAEL J. WOZNY
Department of Electrical and Systems Engineering
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

The effective use of microprocessors in control system design requires a thorough understanding of the trade-offs and limitations of hardware, software and sampling theory on feedback design techniques. This paper describes the laboratory program at RPI which provides the control engineer with the basic lab experience in this important area.

INTRODUCTION

Microprocessor applications are different than large computer applications, because they are concerned with real time control signals and not with paper output. Microprocessor programming stresses program size, real time constraints, and peripheral device control. Furthermore, less software support is generally available for microprocessors, which implies increased difficulty in testing and debugging programs. In essence, microprocessor applications are concerned with hardware systems for some particular use.

The RPI laboratory program in the microprocessor applications area reflects this viewpoint. Three parallel laboratories are offered: 1) Introduction to Microprocessors, 2) Computer Applications Laboratory, and 3) Computer Hardware Design. (Microprocessor experiments are also taught as part of other laboratories.) The first lab stresses software and a general understanding of how microprocessors work. The lab/course is populated with students from a variety of disciplines. The second lab uses the microprocessor as an element in a larger system. The use of the microprocessor is dictated entirely by the application, such as sampling times, ...etc. The third lab emphasizes hardware interfacing. All real microprocessor applications must eventually involve this consideration.

Control engineers at RPI are free to take all three labs, depending on the degree of interest in microprocessors. Each lab is discussed in detail in this paper.

INTRODUCTION TO MICROPROCESSOR LAB

This laboratory uses the Motorola M6800 microprocessor evaluator kit. The lab was developed by the author. The early experiments in the lab provide the student with experience in writing machine code programs and in using the debugging features in the microprocessor. The next set of experiments require the student to write device handlers for the keypad and the 7-segment LED displays. These handlers are then used in a subsequent experiment where the displays are time shared at a rate determined by the keypad input. Serial and parallel I/O are studied next, including the handling of interrupts. This leads to the problem of downloading one microprocessor from another via a parallel interface. Both microprocessors are busy until the master unit needs data. The slave unit is then interrupted and the necessary data sent to the master unit, where it is processed. Both units continue operating as before. The final experiments illustrate program development procedures using the Motorola Exorcisor and specific applications such as traffic controllers.

Approximately 100 students enroll in this course each year. Ten stations of each experiment are provided. Many control engineers take this course for additional background in microprocessors.

COMPUTER APPLICATIONS LAB

This laboratory was developed by Profs. H. Kaufman and L. Gerhardt about three years ago. The goal of the lab is to develop the student's ability to define and solve realistic engineering problems arising in the applications of computers to automatic control and communication systems. The students are given large (50 page) hand-outs for each experiment which cover the theory of the particular application and instructions for evaluating the design. Only a few lectures are given to supplement the lab. The student must study the theory, carry out the design and evaluate the results on his own. Each experiment is the result of a Master's thesis in the past.

The laboratory uses the IMSAI 8080A microcomputer with assorted I/O devices (keypad, 7-segment LED displays, binary LED outputs, binary switch inputs, A/D and D/A converters) and 4K bytes of memory. The software required to run each experiment resides in PROM which is incased in a cartridge which plugs into the back of the microcomputer through an edge connector. These programs, written in machine code for maximum efficiency, perform the specific computation and data transfer operations required for each experiment. Data for these programs is supplied by the student via the I/O devices as required by the experiment. Oscilloscopes, function generators, strip chart recorders as well as motors and analog computers are available for particular experiments. Microcomputer peripherals for development work include an LSI CRT terminal, a dual floppy disk drive and a Potter line printer.

The experiments for this lab are

- Familiarization with equipment
- Design of analog and digital compensators for a DC motor feedback system
- Linear optimal control
- Design of PID controllers for process applications
- Digital voice processing and delta modulation
- Binary communications
- Digital filtering
- Digital logic

DC Motor Experiment

In the DC motor experiment, the student is given the motor transfer function and he is required to design (1) an analog compensator, (2) a digital approximation to the analog compensator via difference equations and z-transforms, and (3) digital controllers via minimal prototype design criteria and also ripple free response, according to specifications on overshoot, settling time, and dead zone in output. The Feedback Limited motor control unit ES130 is used.

Optimal Control Experiment

In the optimal control experiment, a given quadratic performance criteria is minimized and the Riccati equation solved to determine the feedback gains for the controller. The plant is simulated on the analog computer and the controller implemented on the digital computer. A/D and D/A converters are used to make the necessary analog-digital conversions.

Process Control Experiment

The plant is simulated on the analog computer in the Process control experiment also. Various digital PID controller designs based on Ziegler-Nichols, Gallier-Otto, Graham-Lathrop, and Finite settling time, ripple free schemes are studied along with the effects of sampling time and delay time. The controllers are compared for rise time, overshoot, settling time and steady state error.

COMPUTER HARDWARE DESIGN LABORATORY

The microcomputer used in this lab is the DEC LSI-11. Students first learn how to use high frequency instrumentation and then study the transmission line effects associated with high speed logic. After a study of various logic families, they are required to design and build a controller for a floppy disk and for a direct memory access channel. This lab gives the student the necessary hands on experience to develop confidence in hardware interface design.

This course and the associated lab was developed by Prof. J. McDonald. He also offers an advanced hardware course which deals with bit slice technology, microprogramming and with very high speed logic design. All these areas are becoming more important to the control engineer.

SUMMARY

The microprocessor is rapidly becoming the single most important element in control system design. To provide the necessary lab experience in this area RPI offers three parallel microprocessor laboratories for Junior/Senior students.

These laboratories acquaint the control engineer with real time on-line applications as well providing the opportunity to become involved with computer hardware, software and overall sampled data system design involving the computer as an elemental block.

A SENIOR-LEVEL COURSE ON MICROPROCESSOR BASED CONTROL SYSTEMS

F. A. BRIGGS and A. J. KOIVO
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

This paper discusses an undergraduate control systems curriculum that leads to a microprocessor-based control systems course for seniors whereby the computer and control backgrounds of students are merged. The course requires knowledge in control systems as well as computer hardware and software.

INTRODUCTION

Since the advent of microprocessors, the design of many control systems has been evolving around microcomputers. The incorporation of microcomputers in industrial applications has enhanced productivity and reliability by adding local "intelligence" to many industrial machines. Microprocessors were easily accepted by system designers because they possess cheap digital processing capability and can replace a subset of digital systems. Other attributes of microprocessors that make them so attractive to control systems applications are their portability and adaptability. Today, we find their uses in such control systems as robots, missiles, and automobiles.

Although the application of microprocessors to control systems is gaining in popularity, systems designer should learn to evaluate the applicability of a particular microprocessor to a design problem. For example, digital controllers are prone to round-off errors due to quantization of system variables. Their insensitivity to noise and thermal drift may make them desirable in many applications. Furthermore, their decision making capability and reprogrammability often make them attractive in other system applications.

One of the major problems in the application of microprocessors to control systems is in the speed of the microprocessor. Current single-chip microprocessors are too slow for many real-time control applications in which the response time is required to be small. In other applications which require high sampling rate or more complex control systems, the conventional microprocessor is often inappropriate. There are two approaches to alleviate the common limitations of a conventional microprocessor: one is to use faster processors, such as bit slice microprocessors, and the other by multiprocessing. The current trend in implementing more complex controllers is to use distributed microprocessors to enhance the computational power of the system.

The undergraduate curriculum in control systems at Purdue University is attempting to prepare graduating students for more complex computer-based control designs by providing a strong background in microprocessor-based control systems at the undergraduate level. The dichotomy of training between the control systems and the computer systems groups are beginning to merge. The control systems engineering graduates of today and the future should have adequate training in the areas of both computers (hardware and software) and controls. We feel that it is time to merge at least the introductory subject material into a single course at the senior level of the undergraduate curriculum.

BACKGROUND ON SYSTEMS ENGINEERING

An undergraduate curriculum suitable for training students in systems engineering with a strong emphasis on digital computer use and applications has been recently proposed to the School of Electrical Engineering at Purdue University. After a sequence of general courses in electrical engineering, circuits and computers (EE 201, 202, 207, 266, 267) the junior year student will be introduced to EE 303 (which is expected to replace the current EE 301) which is one of the basic courses in system theory. Methods of frequency domain system analysis including the use of Fourier and Laplace and z-transforms, Bode plots, stability criteria and other similar techniques are introduced. The next course, EE 382 is mainly a frequency domain course on classical methods for controls of continuous-time systems. Bode plots, Nyquist criterion, and the Root Locus are used for design purposes. Its complement EE 383 (replacing the current EE 483) is proposed to take up the design of discrete time systems. However, since the purpose of systems education at Purdue is the integration of computers and systems, they both lead to a lab EE 484 introducing microprocessors in control.

EE 308 is the lab associated with EE 382 in which several experiments are designed to train a student in the analysis and synthesis of actual control hardware.

Two dual level courses EE 580 and EE 589, available to both qualified undergraduate and graduate students are presently tested to disseminate various optimization methods and estimation and identification techniques, respectively for continuous and discrete time systems. They both heavily use digital computers for problem solving and simulation purposes.

In conclusion the systems program at the School of Electrical Engineering at Purdue University relies heavily on computer education and technology and is already engulfed in the concepts and use of microprocessors.

BACKGROUND ON COMPUTER ENGINEERING

The undergraduate curriculum in computer engineering provides three basic courses in digital hardware and microcomputers which are taken by most students in systems engineering. These three courses have companion laboratories. The first course, EE 266, an Introduction to Digital Logic Design, is taken primarily by freshman and sophomores. This course covers three main subject areas: Boolean Algebra and Combinational Logic Design, Flipflops and Segnential Logic Design, and Binary Counters and Register Design. The companion laboratory course, EE 267, Introductory Digital Subsystems Laboratory, exposes the students to the breadboarding, diagnosis and testing of digital logic systems. The second course, EE 362, Introduction to Computer Engineering, is intended to introduce junior level students to assembly language programming concepts and the hardware aspects of microprocessor, using the Intel 8080/8085 microprocessor. The hardware aspects of this course builds upon the digital logic background obtained in EE 266, 267. The students learn to interface external devices, such as analog and digital devices, using A/D and D/A large-scale integrated chips through serial and parallel ports. a wide variety of laboratory experiments aid the students in designing interfaces and programming device drivers. In addition to the input/output interfaces and programming, the course introduces advanced programming with basic data structures, and elementary sorting and searching techniques.

The third course taken by students interested in more background in computer systems is EE 466, An Introduction to the Design of Digital Computers. This course taken primarily by seniors, deals with micro- and minicomputer organization with emphasis on the internal computer hardware structure. In the companion laboratory course, EE 467, A Digital Hardware Laboratory, students study basic computer systems using the Motorola 6800 microprocessor as the main processor. Figure 1 shows the sequence of courses that lead to the microprocessor based control course, EE 484.

The student who has taken the sequence of three computer engineering courses is in a position to handle the hardware (interfacing) and software problems encountered in designing microprocessor based control systems.

MICROPROCESSOR BASED CONTROL SYSTEMS LABORATORY COURSE

The course is intended for the students who has already taken a course on microprocessors and on the analysis and design of digital (discrete-time) control systems. Most of the laboratory experiments are based on the material covered in the course on the digital control systems (EE 383). The experiments demonstrate to the students the use of microprocessors in the control systems and, particularly, the implementation aspects of the controller design.

The first part of the course reviews some basic components of control systems. The students are also asked to use Motorola 6800 microprocessors in some simple problems. This is necessary because the students have taken a course on microprocessors in which they have used 8080 (Intel) microprocessors. This part includes general discussions on microprocessor control applications, interrupt processing on real-time control systems, input/output system hardware and I/O programming. Our estimate on the time needed for this part is 6-7 weeks. We hope that in the future we will be able to make this part of the course shorter.

The Intel 8080 would have been a more natural microprocessor to use in the microprocessor based control course since most of the students have had experience with it. However, the system used in EE 362 for "hands on" laboratory is the Intel Prompt 80 microcomputer system which has limited amount of memory and for which new addition of memory is not practical. The Prompt 80 is a stand-alone PROM based monitor system with extensive debugging aids which makes it quite suitable to be used in the first course on microprocessors. We have selected the SWTP 6800 microcomputer system which has the capability to accept add-on memory of up to 32K words required for many of the computations performed in the control system experiments. The source program files are created and edited on a PDP 11/70 Unix-based time sharing systems in the School of Electrical Engineering. A 6800 macro assembler which also exists on the 11/70 is used to assemble the source program files and the object files are down-loaded into the SWTP 6800 microcomputer systems for run-time debugging and testing.

With the appropriate basic knowledge on the use of the microprocessors 6800, the sampling operation, A/D and D/A conversion is presented with detail hardware as well as software. The students are asked to use the A/D and D/A converters in conjunction with specific functions, and to write the software for this experiment.

One of the first complete systems studied by the students is a static optimization problem, in which an RLC-circuit is automatically tuned by a microprocessor. A numerically controlled positional servo motor is used to transfer a screw driver, which is attached to a stepper motor, so that the screw driver will be aligned with the shaft of a capacitor. Then the microprocessor energizes a solenoid moving the screw driver into a slot of the shaft. The RLC-circuit will then be tuned to a specific frequency (as determined by an oscillator) by the microprocessor. This is accomplished by asking the microprocessor to compute a gradient (tangent) at the points on Bode's magnitude graph, until the peak value is located. For the experiment, the students are given the components which they connect appropriately to set up the system. The students will then proceed to write the program for the microprocessor, and test the operation of the system.

The use of a microprocessor as a controller is demonstrated by modelling the plant to be controlled on an analog computer. The students are asked to design a deadbeat controller for a given feedback sampled data control system when a step input is applied to the system. The realization of the controller is performed on the microprocessor. The students are also asked to interface the various components and write appropriate software for the microprocessor.

Our intention is to include into the controller design the use of a feedback controller design (the LPQ-problem) as an application of the dynamic programming method. This particular experiment is currently being investigated in regard to the mathematical operations and microprocessor memory.

Most of the experiments have been constructed as senior projects. The running of the laboratory as a class is under way. Therefore, our experiences on this laboratory course as a whole are quite limited at the present.

CONCLUSION

We have presented the undergraduate curriculum in control systems engineering in the School of Electrical Engineering at Purdue that leads to the basic knowledge on microprocessor based control systems. We hope that each semester, about one third of the laboratory assignments will be rewritten or replaced. The new material will be introduced during the current offering of the course so that the course content does not become stagnant and will permit complete turnover of experiments on the long run.

ACKNOWLEDGEMENT

The authors would like to thank Professor George N. Saridis for his contributions in this paper.

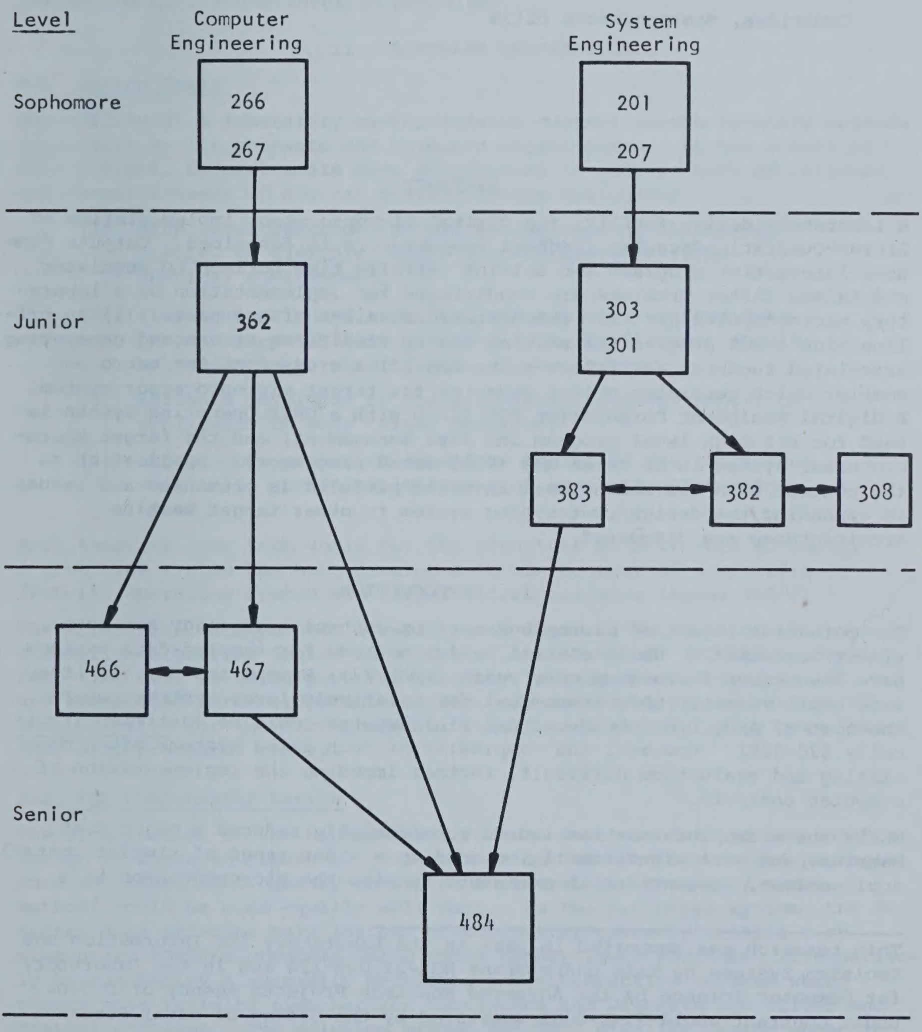

Level
Computer Engineering
System Engineering
Sophomore
266
267
201
207
Junior
362
303
301
383
382
308
Senior
466
467
484

Figure 1. Microprocessor Based Control Systems Course and prerequisite structure.

A USER ORIENTED MICROCOMPUTER FACILITY FOR DESIGNING LINEAR QUADRATIC GAUSSIAN FEEDBACK COMPENSATORS*

PAUL K. HOUPT*,**
JAWED WAHID***
TIMOTHY L. JOHNSON*,**
STEVEN A. WARD*,***

*M.I.T. Department of Electrical Engineering and Computer Science
**M.I.T. Laboratory for Information and Decision Systems
*** M.I.T. Laboratory for Computer Science

Cambridge, Massachusetts 02139

ABSTRACT

A laboratory design facility for digital microprocessor implementation of Linear-Quadratic-Gaussian feedback compensators is described. Outputs from user interactive programs for solving infinite time horizon LQ regulator and Kalman filter problems are conditioned for implementation on a laboratory microcomputer system. The software consists of two parts: (1) an off-line high-level program for solving the LQ Ricatti equations and generating associated feedback and filter gains and (2) a cross compiler/macro assembler which generates object code for the target microprocessor system. A digital Equipment Corporation PDP 11/70 with a UNIX Operating System is used for all high level program and data management, and the target microprocessor system is an Intel MDS (8080-based processor). Application to the control of a two dimensional inverted pendulum is presented and issues in expanding the design/prototyping system to other target machine architectures are discussed.

I. INTRODUCTION

The potential impact of microprocessors on control technology has been widely acclaimed. While control design methods for sampled-data systems have been known for a good many years (Kuo, (1); Bryson and Ho, (2)), they were until recently only economical for relatively large systems, where the cost of A/D, D/A interfaces and minicomputer could be justified (typically $20-30K). However, the complexity of such large systems often made testing and evaluation difficult, further impeding the implementation of computer controls.

While the microprocessor has indeed substantially reduced a major cost barrier, and more signfiicantly, opened up a wider range of simpler control applications, we consider it premature to view the microprocessor as a

---

This research was supported in part in the Laboratory for Information and Decision Systems by NASA under Grant NGL-22-009-124 and in the Laboratory for Computer Science by the Advanced Research Projects Agency of D.O.D. under Contract N0014-75-C-0061 monitored by the Office of Naval Research

panecea for the experienced control engineer, let alone the college or graduate student. Ironically, the microprocessor has presented new barriers to the control designer: assembly-language programming, coding for real-time execution, and many other hardware synchronization problems. These problems are by no means insurmountable but in our opinion they are much alleviated by a development system involving a larger machine with higher level language and multiprocessing capability.

Our purpose herein will be to describe a prototype microprocessor design facility and to communicate the key insights which have been gained thus far in its development. The facility consists of high and low-level software, computer hardware, and a continuous target application involving a two degree of freedom inverted pendulum.

II. SOFTWARE DESCRIPTION

2.1 Design Goals

The utility of a laboratory microprocessor control design facility depends critically on its software and hardware organization. At the outset of this project, several goals were established to address both educational and research needs of digital control system designers:

(1) High level languages should be employed at all design levels, in flexible, multi user-interactive (time-sharing) environment;

(2) A control oriented symbolic language for required data processing activities (control laws and filters) should be available to simplify use of the system;

(3) Software developed in the high level language should be applicable to many target machines;

(4) Object code generated should be optimized for the target machine to permit exploitation of available processing speed

With these as long term goals for the structure of desirable software, a prototype system has been developed in the context of a specific (PDP-11) operating system and target microprocessors (Intel 8080).

The MIT System incorporates several novel features which allow the user to accomplish algorithm design in a higher-level language (FORTRAN), develop real-time code in using powerful assembly-language macro commands, which are translated into relocatable object code and finally cross-assembled automatically for the microprocessor itself, and automatically down-loaded into microprocessor memory.

2.2 LQG Compensator Design

The organization of the higher-level Fortran routines for LQG compensator design is shown in Figure 1. Note that other alternative design packages, such as classical frequency-domain or multivariable frequency domain methods could be used equally well here. In the prototype system, the designer does off-line calculations of controller gains by writing very elementary FORTRAN programs using these higher-level routines; alternatively he can employ a specially developed user-interactive program which covers most infinite time horizon continuous and discrete LQG filter and control problems. The solution method for this off-line design phase in the prototype system employs fairly standard eigenvalue decomposition techniques which were developed at the MIT Electronic Systems Laboratory (3). These FORTRAN programs in turn depend on the "EISPAK" eigenvalue/

eigenvector package developed at Argonne Laboratory; this is widely available. User developed or available interactive programs for post-processing of control and filter designs (including simulation) are useful before proceeding to generate code for the target microcomputer systems.

2.3 Code Generator for Target Micro Computer System

After completing the off-line design phase described in 2.2, code to implement the control law and/or filter algorithm on the target micro computer system is generated following the procedure outlined in Figure 2. This activity consists of three main parts: (1) Generation of an assembly program using a collection of previously defined control-specific macros (MACGEN Figure 3); (2) Cross assembly of the user generated program (a source collection of macros) into executable absolute object code (DOMIC Figure 4); (3) down loading of the absolute code into the target system. The DOMIC program performs the steps illustrated in Figure 4 which includes binding of the relocatable user object files from the (MICAL) cross assembly with the previously defined control macros, and conversion of the resulting files into absolute (executable) code for the target microcomputer system. The tedium of performing these procedures manually is thus eliminated for the user. Note that the user-defined macro source program written with MACGEN is canonical, in that by use of a different cross assembler in MICAL, other target processors can be employed. For example, in our prototype system, both Intel 8080 and Motorola 6800 processors can be employed as the target system. Furthermore, the use of special architectual units such as hardware multiply, array processors, etc. can be made essentially invisible to the control designer since the special needs of these units (or even I/O devices) are encoded in the library of MACGEN routines. All program generation in the above system is handled under the supervision of the host operating system, which in the MIT prototype was "UNIX" as implemented on a DEC PDP 11/70. The powerful text editing and file handling capabilities of such systems (or similar) are, in our view, essential in an educational or research environment so that the user (student or researcher) can concentrate on the control/estimation aspects of the problem.

2.4 A Simple Example

It would be impossible in this space to enumerate all the capabilities of the current system implemented at MIT. We will be content to illustrate the heart of the software, the MACGEN program, with a simple Example. A program for a state regulator with (second-order) dynamics (one-input and two-outputs) is illustrated in Figure 5. The general flow of this program is illustrated Figure 6, which also illustrates some additional special features in our prototype system. The processor is assumed to be interrupt driven by an external clock. At each sample time the interrupt routine (INT5:) is serviced. A/D conversions are performed setting

$$x_1 = \text{A/D PORT } 0$$

$$x_2 = \text{A/D PORT } 1$$

Then

$$u = g_1 x_1 + g_2 x_2$$

is output through port 0 of the D/A converter; the elements (g_1, g_2) of "gain" are set in the routine called "ready". The real time operation of this program as it resides in the target machine is explained with the aid of Figure 5a. When the hardware is initialized, the program sets up specified conditions on I/O devices and fetches the required gain data and puts it in a table. A real time monitor program (READY) is then entered which can accept simple commands from a terminal to run, quit, change the gains, etc., all on-line. The control up-dates are accomplished by an external (prioritized) clock interrupt, which causes the branch to "INT5". The program then returns to the READY monitor program.

This example illustrates the relative simplicity of using the mnemonic macros to generate code control updating for real-time (interrupt-driven) operation. More complex control laws including filters, observers, or classical lead-lag compensators are easily constructed with the currently available collection of macros.

III. HARDWARE

3.1 Prototype System

The prototype software described above was implemented as shown in Figure 7 on a DEC PDP 11/70 with 64K of memory, in conjunction with an Intel 8080 based MDS-800 (Development) as the target micro computer system. The MDS system employed hardware multiply, 16K of memory and was outfitted with the Datel D/A and A/D cards shown. (32 A/D inputs, 16 A/D outputs). Two precision D.C. servo amplifiers were provided to drive high power loads (up to 10A. @ 28V.) for simple control designs. A number of custom software modules were written or adapted from available software to handle communications between the MDS monitor and UNIX at the 9.6 K-band rate (including down-load and plotting of hardware results). We believe that our system is well above the minimal "threshold" to support the design software, although no attempt was made to define what lower bound on hardware would suffice. One the other-hand, more sophisticated target systems could have been employed.

3.2 An Application

The prototype system was applied to the design of a two degree of freedom inverted pendulum. The pendulum was a 60" long aluminum pole mounted on a gimbal as shown in Figure 8. The gimbal is mounted to a large (48" x 48") drafting-type X-Y plotter. It consists of a cart which can be translated along an aluminum rail to obtain "X" displacement, and the rail in turn can be translated to obtain "Y" displacements. Motion is achieved with a system of cables and pulleys connected to two 28 volt D.C. servo-motors which can be driven with the amplifier outputs from the system described in 3.1. Linearized analysis of the one-dimensional inverted pendulum is well known and will not be repeated here. Extension to the two dimensional case is straightforward and the reader is referred to Wahid (4) for details. The state variables for the pendulum are:

$$\underline{x} = (x,\dot{x},\theta_x,\dot{\theta}_x,y,\dot{y},\ \theta_y,\dot{\theta}_y)$$

where the angles and relative x,y coordinates are defined in Figure 8. The unforced linearized dynamics decouple between the (x,θ_x) and (y,θ_y) coordinates around $\underline{x} = \underline{0}$. Four observations are available via potentiometers:

$$y_1 = \theta_x$$

$$y_2 = x + .36y$$

$$y_3 = y$$

$$y_4 = \theta_y$$

(The coupling between y_2 and y_3 results from the geometry of the cable/potentiometer connections). An LQ (full-state feedback) regulator was designed as described in Wahid (4) (Figure 9). Rate variables were obtained by digital "lead-lag" software (direct differencing with high-frequency rolloff to prevent noise aliasing). This approach was adopted so that sufficiently fast update rates could be achieved with a single 8080 C.P.U. A full-order Kalman filter or observer, could not be up-dated fast enough with the 8080. Reduced order compensators, use of hardware multiply and two (parallel) processor implementations are currently being investigated with the single C.P.U. design. The pendulum was easily stabilized with the LQ gains obtained using an eighth-order dynamic model.

IV. CONCLUSIONS

The prototype control system design tool described has served to meet at least some of the goals outlined in Section 2. Where larger time sharing systems are present, interactive high-level languages for control algorithm design, simulation, and laboratory evaluation provide a favorable environment for both educational and research needs. The essence of our approach is the use of a library of macros for performing a broad spectrum of common control activities. The user can tailor these program modules as his design needs dictate, with code that is relatively easy to use and debug. A second generation of the MACGEN language is currently planned which will provide an even more transparent collection of mnemonics for the required signal processing activities.

V. REFERENCES

[1] Kuo, B.C., Discrete Data Control Systems, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1970.

[2] Bryson, A.E., and Ho, Y.-C., Applied Optimal Control, Blaisdell Publishing, Waltham, Mass., 1969.

[3] Sandell, N.R. and M. Athans, Fortran Computer Routines for the Solution of Linear Quadratic Gaussian Optimal Control Problems, MIT Center for Advanced Engineering Study, Cambridge, Mass., 1974.

[4] Wahid, J., "Implementation of Linear Quadratic Gaussian Compensator on Microprocessors", S.M.E.E. Thesis, MIT Department of Electrical Engineering and Computer Science, May 1978.

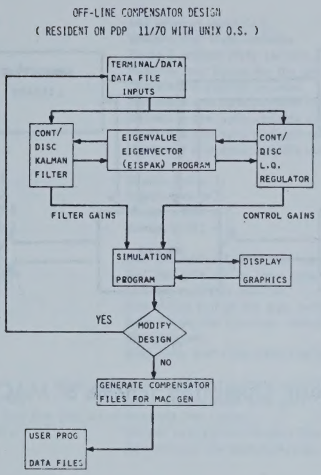

OFF-LINE COMPENSATOR DESIGN
(RESIDENT ON PDP 11/70 WITH UNIX O.S.)
TERMINAL/DATA DATA FILE INPUTS
CONT/ DISC KALMAN FILTER
EIGENVALUE EIGENVECTOR (EISPAK) PROGRAM
CONT/ DISC L.Q. REGULATOR
FILTER GAINS
CONTROL GAINS
SIMULATION PROGRAM
DISPLAY GRAPHICS
YES
MODIFY DESIGN
NO
GENERATE COMPENSATOR FILES FOR MAC GEN
USER PROG DATA FILES

Fig. 1 Interactive(FORTRAN) Compensator Design Program

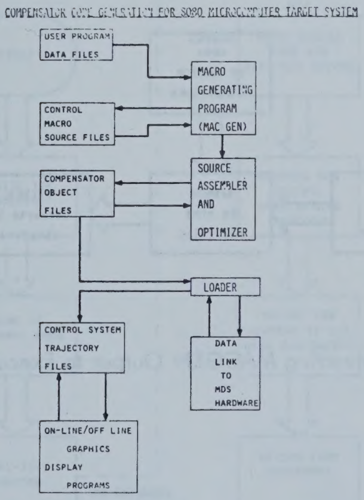

COMPENSATOR CODE GENERATION FOR 8080 MICROCOMPUTER TARGET SYSTEM
USER PROGRAM DATA FILES
MACRO GENERATING PROGRAM (MAC GEN)
CONTROL MACRO SOURCE FILES
COMPENSATOR OBJECT FILES
SOURCE ASSEMBLER AND OPTIMIZER
LOADER
CONTROL SYSTEM TRAJECTORY FILES
DATA LINK TO MDS HARDWARE
ON-LINE/OFF LINE GRAPHICS DISPLAY PROGRAMS

Fig. 2 Compensator Code Generation for Target System

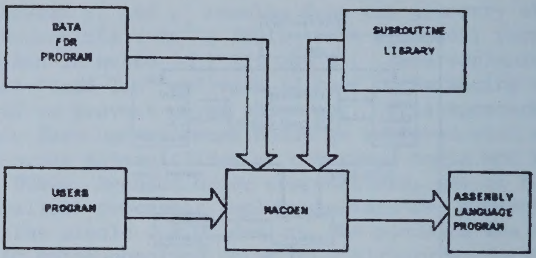

DATA
FOR
PROGRAM
SUBROUTINE
LIBRARY
USERS
PROGRAM
MACGEN
ASSEMBLY
LANGUAGE
PROGRAM

Fig. 3 Basic Operating Scheme of MACGEN

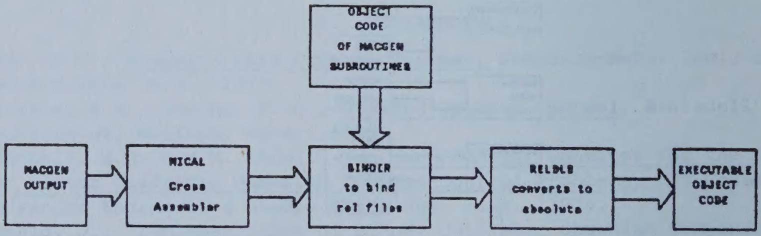

OBJECT
CODE
OF MACGEN
SUBROUTINES
MACGEN
OUTPUT
MICAL
Cross
Assembler
BINDER
to bind
rel files
RELDLD
converts to
absolute
EXECUTABLE
OBJECT
CODE

Fig. 4 Converting MACGEN Output to Executable Form

```
         .globl int5                   |set interrpt level to 5.
         .var x1                       |declare first state variable.
         .var x2                       |declare second state variable.
         .varp gain  4                 |reserve four bytes for the gains.
         .var u                        |declare the control variable.
         .var one 1                    |declare a one byte variable called one.
         .var two 1                    |declare a one byte variable called two.
         .var port0 1                  |declare a one byte variable called port0.
         .var port1 1                  |declare a one byte variable called port1.

         .asgn one 1                   |assign one - 1.
         .asgn two 2                   |assign two - 2.
         .asgn port0 0                 |assign port0 - 0.
         .asgn port1 1                 |assign port1 - 1.

         init                          |call the system initializtion routine.
         date two gain                 |get two gain values from data.
         ready                         |call the real time monitor.
                                       |this is the end of the main program loop.
                                       |now comes the interrupt driven portion of
                                       |the program.
                                       |that does that computation of the control.

int5:                    atods port0 x1|input first state from port 0 of A/D.
         atods port1 x2                |input second state from port 1 of A/D.
         mvmul one two one gain x1 u   |compute the control.
         dtoas port0 u                 |output new control to port 0 of D/A.
         return                        |return from the interrupt loop.
```

Fig. 5 Sample MACGEN Program Implementing a Second Order State Regulator

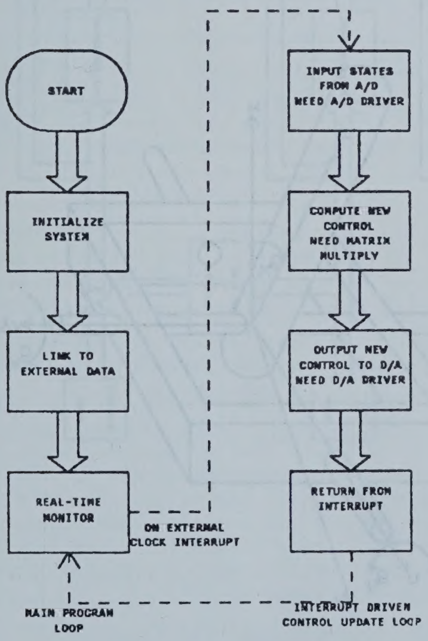
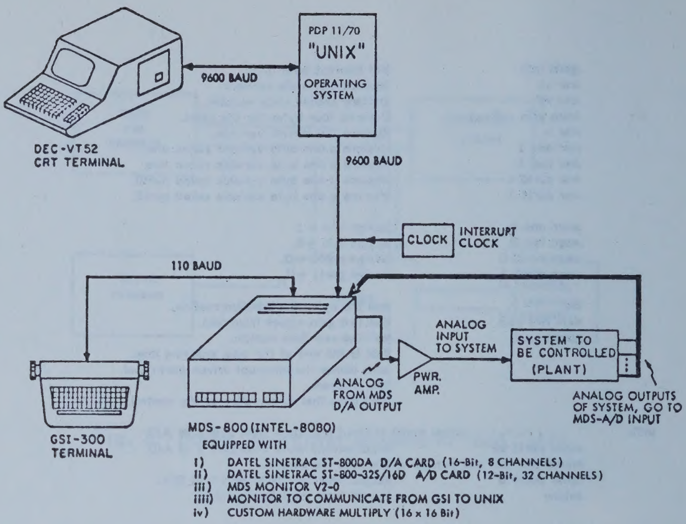

START
INITIALIZE SYSTEM
LINK TO EXTERNAL DATA
REAL-TIME MONITOR
ON EXTERNAL CLOCK INTERRUPT
MAIN PROGRAM LOOP
INPUT STATES FROM A/D NEED A/D DRIVER
COMPUTE NEW CONTROL NEED MATRIX MULTIPLY
OUTPUT NEW CONTROL TO D/A NEED D/A DRIVER
RETURN FROM INTERRUPT
INTERRUPT DRIVEN CONTROL UPDATE LOOP

Fig. 6 On-line Computation Required by State Feedback Regulator


PDP 11/70
"UNIX"
OPERATING SYSTEM
9600 BAUD
DEC-VT52 CRT TERMINAL
9600 BAUD
CLOCK
INTERRUPT CLOCK
110 BAUD
ANALOG INPUT TO SYSTEM
SYSTEM TO BE CONTROLLED (PLANT)
ANALOG FROM MDS D/A OUTPUT
PWR. AMP.
ANALOG OUTPUTS OF SYSTEM, GO TO MDS-A/D INPUT
GSI-300 TERMINAL

MDS-800 (INTEL-8080)
EQUIPPED WITH

I) DATEL SINETRAC ST-800DA D/A CARD (16-Bit, 8 CHANNELS)
II) DATEL SINETRAC ST-800-32S/16D A/D CARD (12-Bit, 32 CHANNELS)
III) MDS MONITOR V2-0
IIII) MONITOR TO COMMUNICATE FROM GSI TO UNIX
IV) CUSTOM HARDWARE MULTIPLY (16 x 16 Bit)

Fig. 7 Prototype Hardware Realization for 8080 Target System

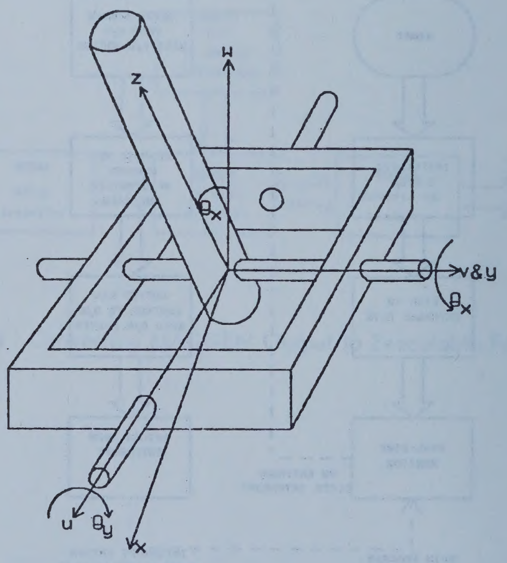

H
Z
θ_x
v&y
θ_x
u
θ_y
x

Fig. 8 Inverted Pendulum and its Gimbal Mounting

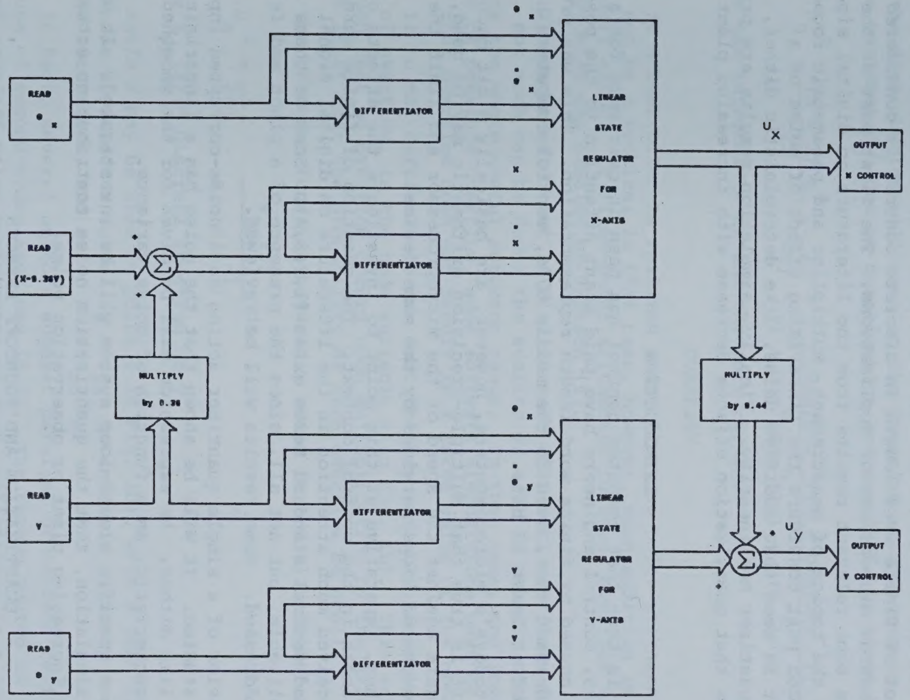

READ
θ x
READ
(X-0.36V)
+
+
MULTIPLY
by 0.36
READ
Y
READ
θ y
DIFFERENTIATOR
DIFFERENTIATOR
DIFFERENTIATOR
DIFFERENTIATOR
θ x
θ̇ x
X
Ẋ
θ x
θ̇ y
Y
Ẏ
LINEAR
STATE
REGULATOR
FOR
X-AXIS
LINEAR
STATE
REGULATOR
FOR
Y-AXIS
U x
OUTPUT
X CONTROL
MULTIPLY
by 0.44
+
U y
+
OUTPUT
Y CONTROL

Fig. 9 Flowchart for the Feedback Compensator using Software Differentiators

EFFECT OF FINITE WORD LENGTH IN DISCRETE-TIME CONTROL

MUSTAFA AHMED and PIERRE R. BELANGER
Department of Electrical Engineering
McGill University
Montreal, Quebec, Canada H3A 2A7

SUMMARY

The effect of finite word length in discrete control is considered, with a view towards microprocessor applications. The first part of the papers reviews some relevant results from the literature on digital signal processing on the topics of measurement, multiplier and parametric roundoff. The second part considers the linearizing effect of noise on a quantizer. It is seen that additive noise, like deterministic dither, smooths the quantizer nonlinearity. Finally, simulation results are presented to show that quantization effects decrease with increasing plant noise.

INTRODUCTION

While it is true that computer control has been in existence for a number of years, control engineers have paid scant attention to the possible problems caused by finite word length representation. This was probably due to the fact that, even in the middle 60's, control computers had word lengths of at least 12 bits.

Many of todays's microcomputers, however, are basically 8 bit machines. While it is true that multiple-precision arithmetic may be used, this may mitigate against the speed of the microprocessor and limit the number of time-shared loops serviced by the same device.

It is, then, imperative at this point to investigate the effects of finite word length in the control context. The question of finite word length has received much attention in the literature on digital signal processing, and several standard texts exist [1,2,3,4]. Some of those results are applicable, but not all, since the presence of a plant and feedback is not addressed. Some results will be reviewed.

The behavior of a single quantizer acting on a noise-corrupted input will then be studied. It will be shown that the noise has a linearizing effect, much like dither. An expression will be given for the smoothed quantizer characteristic, as a function of noise variance.

Finally, a specific closed-loop system will be investigated. It will be shown, by simulation, that the quantization noise contribution actually decreases for increasing plant or observation noise.

QUANTIZATION AND ROUNDOFF EFFECTS

Stochastic modelling of quantization errors is covered in the basic texts, and also in [5] and [6], the latter in a feedback context. Roundoff and quantization errors are modelled as independent, white, uniformly-distributed random variables, with limits $\pm q/2$, If $b + 1$ is the computer word length, then $q = 2^{-b}$. It is easy to show that these random variables have zero mean and a variance of $q^2/12$.

It is assumed that fixed-point arithmetic is used: this implies that there are no roundoff errors in addition and subtraction. As shown in Fig. 1a), the plant has a discrete transfer function $G(z)$ and the controller transfer function is:

$$D(z) = \frac{R(z)}{S(z)} = \frac{r_0 + r_1 z^{-1} + \ldots + r_N z^{-N}}{1 + s_1 z^{-1} + \ldots + s_N z^{-N}} \tag{1}$$

Fig. 1b) shows the A/D quantization error $e\,y$ and the multiplier roundoff errors $e\,r$ and $e\,s$, for a direct realization of the controller. The random variables may be combined into random variables E_1 and E_2, as in Fig. 1c), both with variances $(N + 1)\,q^2/12$. Note that, by the Central Limit Theorem, the distribution will tend to the Gaussian as N increases. The output noise variance σ_0^2, due to quantization and roundoff, is given by

$$\sigma_0^2 = \frac{(N + 1)\,q^2}{12} \cdot \frac{1}{2\pi j} \quad \frac{G(z)G(z^{-1})\,[1 + D(z)D(z^{-1})]}{[1 + G(z^n)D(z^n)]\,[1 + G(z^{-1})D(z^{-1})]}\,dz \tag{2}$$

where the integral is around the unit circle.

OVERFLOW

If a controller is realized using floating-point arithmetic, overflow will not be a problem; in that case, however, each addition is subject to roundoff error. If fixed-point arithmetic is used, overflow is of concern. The problem was studied by Jackson [5].

It can be said that if D(z) is such that u(t) overflows for normal y(t), not much can be done; the scaling of u(t) is usually such that 2^b covers the whole useful actuator range. In that case, and indeed in all cases of possible overflow, some software dispositions must be taken to ensure that overflow does not result in a large value of the opposite sign.

It is not difficult to see that the key variable, from the point of view of overflow, is the node variable x in Fig. 1b). This variable must be such that no overflow occurs upon multiplication by the gains r_i or s_j, or at least that the probability of such overflow be small.

The transfer function between y and x is 1/S(z). Therefore,

$$\sigma_x^2 = \frac{1}{2\pi j} \quad \frac{\phi_{yy}(z)}{S(z)S(z^{-1})}\,dz \tag{3}$$

where $\Phi_{yy}(z)$ = Discrete spectrum of y.

Again K may be used at point 1 in Fig. 1b), with a gain 1/K at point 2. To choose K, select a value $\sigma_x = \sigma_{xd}$ to ensure a suitably small probability of saturation, and set $K = \sigma_{xd}/\sigma_x$.

In fact, several authors suggest parallel or cascade realizations by the use of second-order sections [6,7]. The reason quoted in [6] is the better roundoff noise performance of such realizations in a simulated feedback example. As for [7], the issue there seems to be one of sensitivity of the poles to the coefficients of the characteristic polynomial; this is probably more serious in digital filters, where one often synthesizes high-Q filters, than in control. In any case, Fig. 2 shows a typical second-order section; its use does offer the advantage of having one adjustable overflow gain per section.

COEFFICIENT QUANTIZATION

The effect of coefficient quantization has been treated in [1,8,9]. The result of such quantization is to introduce an error in the value of the coefficient. If floating-point arithmetic is used, this implies a relative error of one part in 2^{b-1} (one of the bits in a sign bit); this represents less than 1% for an 8-bit word. Errors of that magnitude may be serious in high-Q synthesis, but not usually in a control system.

If fixed-point arithmetic is used, however, the error is absolute rather than relative: small coefficients may be quantized out of existence. In any case, the effect on overall system performance may be studied through the use of standard sensitivity analysis [Eq. 10].

QUANTIZER SMOOTHING BY NOISE

We now consider the average response of a single quantizer. It is not difficult to establish that a quantizer may be represented as in Fig. 3. Let the input to the quantizer be

$$x = r + \omega$$

where r is a constant and ω is Gaussian, zero-mean, with variance σ^2. We shall now compute E[y]. We have

$$E[y] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma} \exp(-\frac{\omega^2}{2\sigma^2})\, f(r + \omega)\, d\omega \tag{4}$$

If we let $F(\Omega)$ be the Fourier transform of f(x),

$$E[y] = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\,\sigma} \exp(-\frac{\omega^2}{2\sigma^2})\, \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega) e^{j\Omega(r + \omega)}\, dr\, d\omega \tag{5}$$

Inverting the order of integration and using the known result for the transform of a Gaussian pulse yields

$$E[y] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega) \exp(-\frac{\sigma^2\Omega^2}{2})\, e^{j\Omega r}\, d\Omega$$

$$= \mathfrak{F}^{-1}\, [F(\Omega) \exp(-\frac{\sigma^2\Omega^2}{2})] \tag{6}$$

Note that E[y] is a function of r.

If $f_1(x)$ is f(x) truncated to the principal interval -q/2 to q/2, then

$$F(\Omega) = \sum_{n=-\infty}^{\infty} F_1(\Omega) e^{jn\Omega q}$$

where $F_1(\Omega) = \mathfrak{F}\,[f_1(x)]$

Thus,

$$E[y] = \sum_{n=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{jn\Omega q} \exp(-\frac{\sigma^2\Omega^2}{2}) \int_{-q/2}^{q/2} (-\eta) e^{-j\Omega\eta} d\eta\, e^{j\Omega r}\, d\Omega \tag{7}$$

Inverting the order and using the Gaussian transform,

$$E[y] = \sum_{n=-\infty}^{\infty} \int_{-q/2}^{q/2} (-\eta)\, \frac{1}{\sqrt{2\pi}\,\sigma} \exp[-\frac{(\eta-r-nq)^2}{2\sigma^2}]\, d\eta \tag{8}$$

This may be reduced to

$$E[y] = \sum_{n=-\infty}^{\infty} \left\{ -(r + nq)\left[\mathrm{erf}\left(\frac{-q/2 - r - nq}{\sigma}\right) - \mathrm{erf}\left(\frac{q/2 - r - nq}{\sigma}\right)\right] + \frac{\sigma}{\sqrt{2\pi}}\left[\exp\frac{(q/2 - r - nq)^2}{2\sigma^2} - \exp\frac{(q/2 + r + nq)^2}{2\sigma^2}\right] \right\} \qquad (9)$$

Fig. 4 shows the expected value of the quantizer output ($E[y] + x$), for $\sigma = .2$, 1 and 2; the quantizer step is $q = 1$. For the same conditions, Fig. 5 gives the smoothed nonlinearity in the quantizer model of Fig. 3.

It was surmised that the effect of quantization noise would diminish as the plant noise increased. Assuming $x = r + \omega$, the following crude assumption was made: rather than assume the noise at y in Fig. 3 to be uniformly distributed between $\pm q/2$, we assumed it to be uniformly distributed between the peak values in Fig. 5. The resulting variance, normalized by $q^2/12$, is plotted as crosses in Fig. 6 against the variance (normalized) of ω. Because of the crudity of the assumption, this was tested by a computer experiment. A sampled sine wave $r(t)$ corrupted by additive white noise $\omega(t)$ was passed through a quantizer, the output being $z(t)$. The quantity $\frac{\sigma z^2 - \sigma\omega^2}{q^2/12}$ should correspond to σy^2, the variance at the nonlinearity output in Fig. 3. That quantity is plotted, for sinusoids of periods 8 and 15, in Fig. 6. The agreement with the prediction is seen to be rather good.

A CLOSED-LOOP SIMULATION

Plant equations are given in ARMA form as follows:

$$A(q^{-1})y(t) = B(q^{-1})u(t - 2) + C(q^{-1})e(t) \qquad (10)$$

where

$$A(q^{-1}) = (1 + .95q^{-1})(1 - 1.7q^{-1} + .7q^{-2})$$

$$B(q^{-1}) = 1 - .65q^{-1} + .3234q^{-2}$$

$$C(q^{-1}) = (1 + .95q^{-1})(1 + 1.5q^{-1} + .9q^{-2})$$

$$q^{-1} = \text{unit delay.}$$

The minimum-variance regulator for this plant is [11].

$$S(q^{-1})u(t) = -R(q^{-1})y(t) \qquad (11)$$

where

$$R(q^{-1}) = 5.64 + 3.118q^{-1} - 2.128q^{-2}$$

$$S(q^{-1}) = 1 + 2.55q^{-1} - 2.4034q^{-2} - 1.035q^{-3}$$

The regulator is assumed to be implemented by parallel second-order sections on a computer with an 8 bit word, 1 bit being used for the sign. The quantization step is then $q = 1/2^7 = .0078$.

Figure 7 shows three graphs of output standard deviation σ_y vs the standard deviation σ_e of the white driving noise $e(t)$. The lowest graph represents σ_y in the absence of quantization. The top curve shows σ_y calculated by appending to each multiplication a quantization white noise source, as in Fig. 1. The middle curve is the result of the computer experiment. Note that 1) there is a best value of σ_e, very nearly equal to the quantization step; 2) neither prediction is very good for small σ_e, one being overly optimistic and the other overly pessimistic; 3) the fact that the simulation result approaches the no-quantization results is in line with our results of the previous section.

CONCLUSIONS

In this paper, some results on quantization in digital signal processing have been reviewed and assessed for applicability in a control context. New results, albeit of a preliminary nature, have been presented to show that quantization becomes unimportant in the presence of a moderate amount of noise.

This last result may not be as practically encouraging as it seems. It is often the case that the word length must cover the complete linear range of the measuring instrument. If operation is limited to a small portion of that range near the operating point (as in process control), the noise variance may indeed not be large with respect to the quantization step.

One important topic has been left out: limit cycling. We have obtained quantitative results to show that limit cycles are quenched if sufficient plant noise is present. This has been known in the case of deterministic systems [12], and the use of dither has been suggested. It appears that here also it is beneficial to have some moderate amount of external signal.

REFERENCES

1. L.R. Rabiner and B. Gold, "Theory and application of digital signal processing" Prentice-Hall, 1975.
2. B. Gold and C.M. Rader, "Digital processing of signals" McGraw-Hill, 1969.
3. S.A. Tretter, "Introduction to discrete-time signal processing" Wiley, 1976.
4. A.V. Oppenheim and R.W. Shafer, "Digital signal processing" Prentice-Hall, 1975.
5. L.B. Jackson, "Roundoff noise analysis for fixed point digital filters realized in cascade or parallel form" IEEE Trans. on Audio and Electroacoustics, Vol AU-18, No 2, June 1970.
6. J.B. Knowles and R. Edwards, "Effect of finite-word-length computer in sampled-data feedback systems" Proc. IEEE, Vol 112, No 6, 1965.
7. B. Liu, "Effect of finite word length on the accuracy of digital filters" IEEE Trans. on Circuit Theory, Vol CT-18, No 6, Nov. 1971.
8. C.T. Mullis and R.A. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters" IEEE Trans. on Circuits and Systems, Vol CAS-23, No 9, Sept. 1976.
9. J.B. Knowles and E.M. Olcato, "Coefficient accuracy and digital filter response" IEEE Trans. on Circuit Theory, Vol CT-15, March 1968.
10. R. Tomovic, "Sensitivity analysis of dynamic systems" McGraw-Hill.
11. K.J. Aström, "Introduction to stochastic control theory" Academic Press, 1970.
12. G. Zames and N.A. Schneydor, "Structural stabilization and quenching by dither in nonlinear systems" IEEE Trans. on Aut. Control, Vol AC-22, No 3, June 1977.

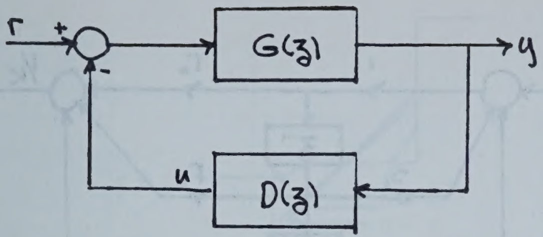


Fig. 1(a) Plant and digital controller

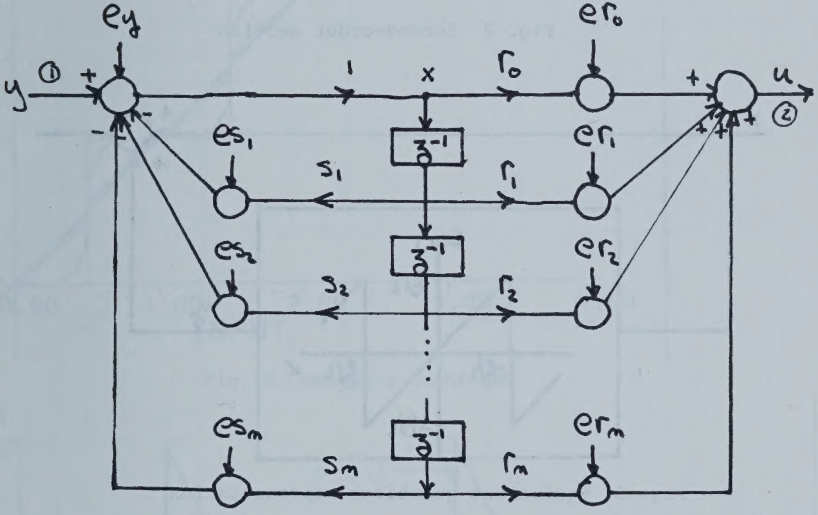

ey
y
①
+
-
-
1
x
r₀
er₀
+
u
②
es₁
z⁻¹
er₁
s₁
r₁
es₂
z⁻¹
er₂
s₂
r₂
esₘ
z⁻¹
erₘ
sₘ
rₘ

Fig. 1(b) Direct realization of regulator

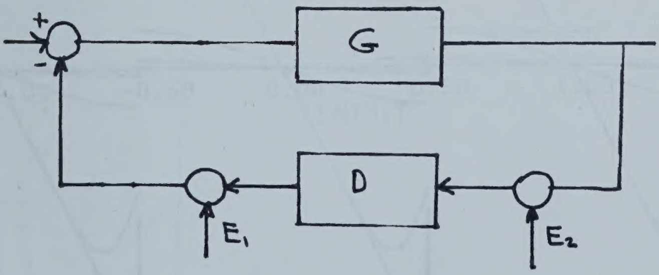


Fig. 1(c) Equivalent quantization noise sources

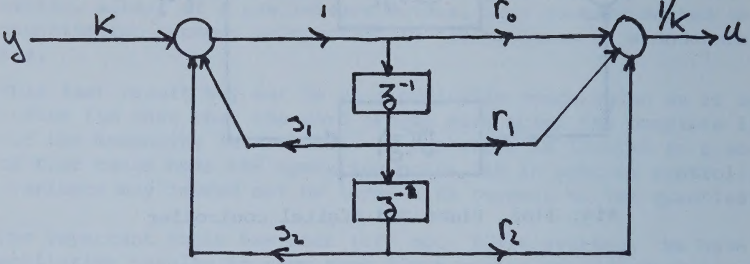

y
K
1
r₀
1/K
u
z⁻¹
z⁻¹
s₁
r₁
s₂
r₂

Fig. 2 Second-order section

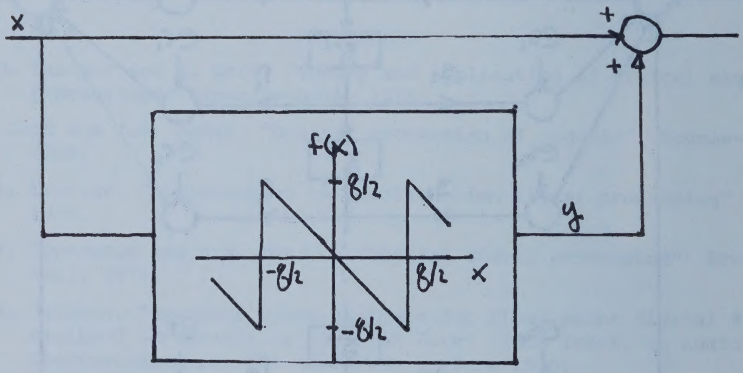

x
+
+
f(x)
δ/2
-δ/2
δ/2
-δ/2
x
y

Fig. 3 Equivalent quantizer representation

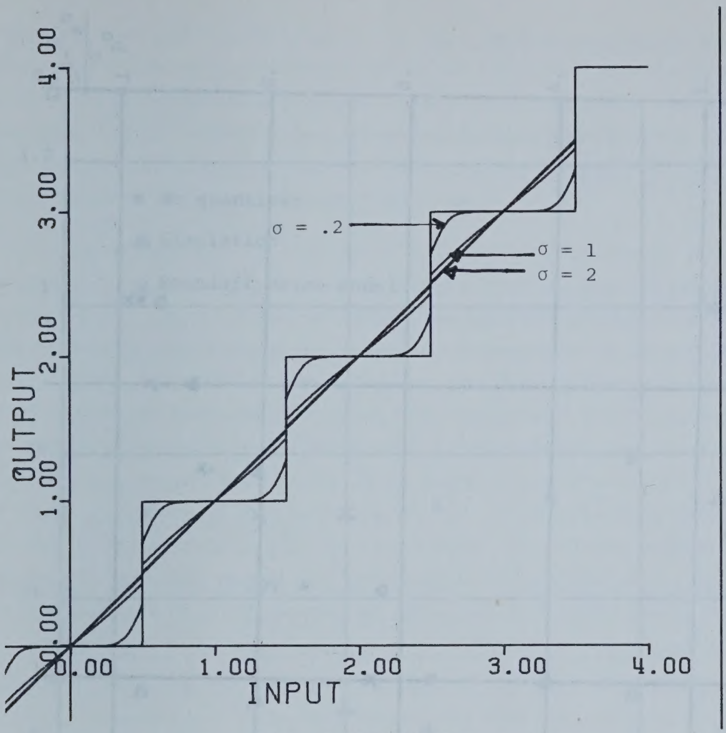


Fig. 4 Smoothed quantizer

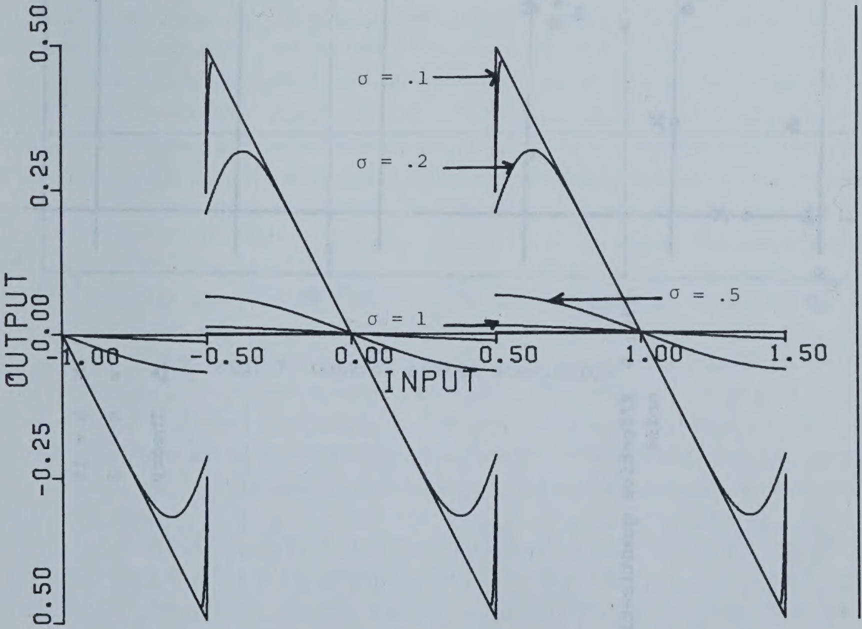


Fig. 5 Smoothed equivalent nonlinearity

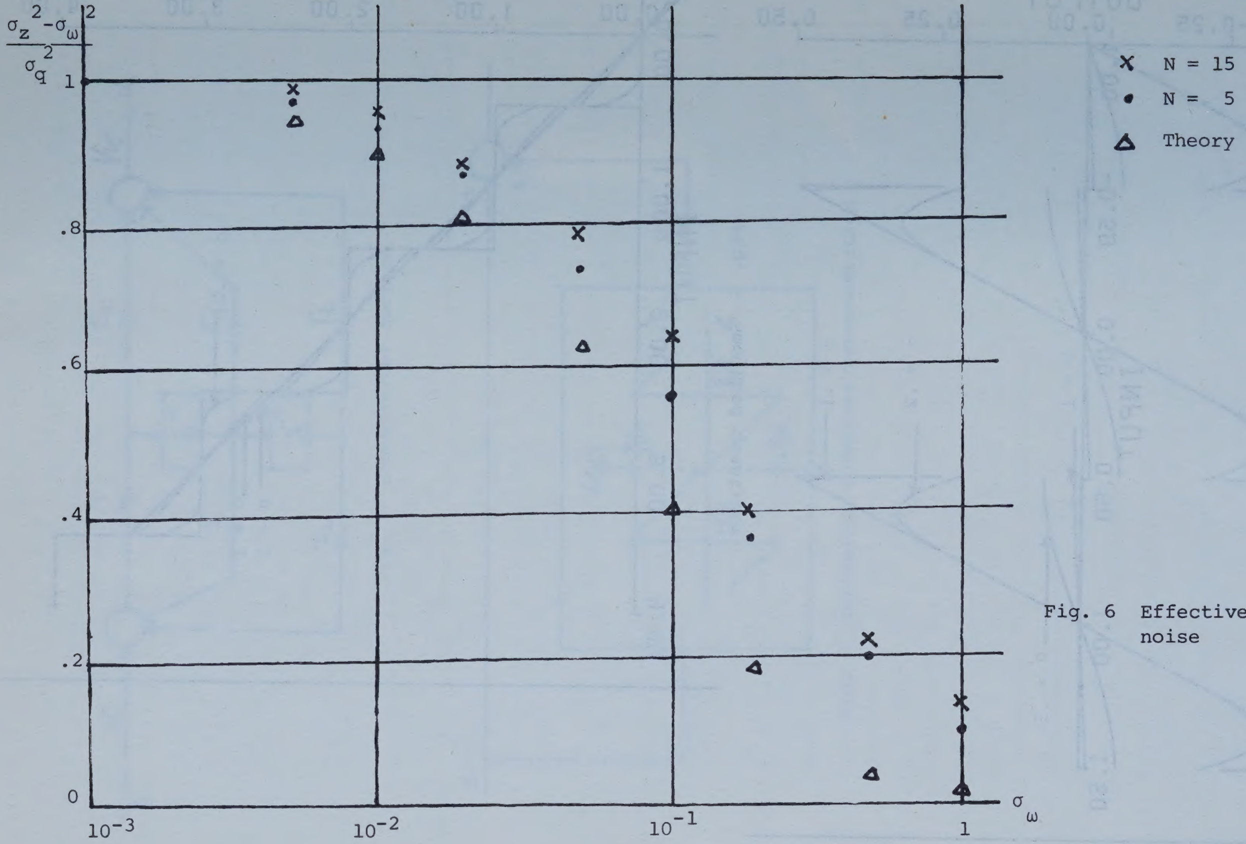


Fig. 6 Effective quantization noise

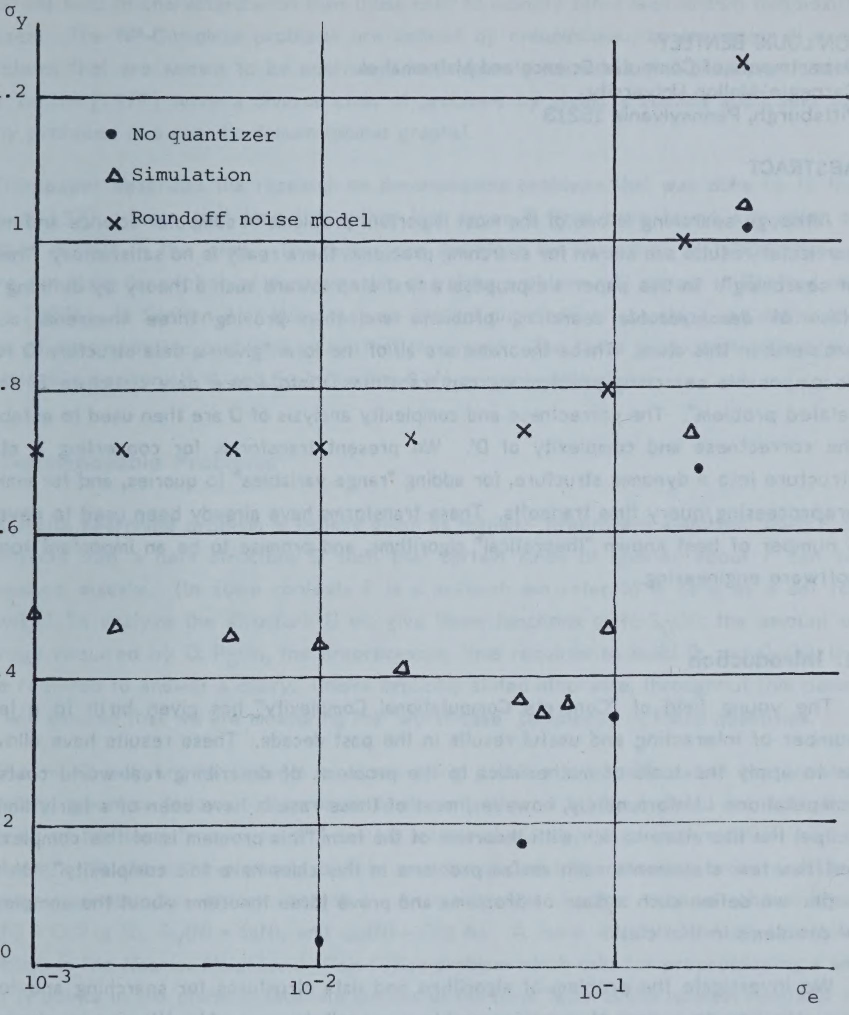


Fig. 7 Control system simulation

DECOMPOSABLE SEARCHING PROBLEMS

JON LOUIS BENTLEY
Departments of Computer Science and Mathematics
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

Although searching is one of the most important problems in computer science and many particular results are known for searching problems, there really is no satisfactory "theory of searching". In this paper we propose a first step toward such a theory by defining the class of *decomposable* searching problems and then proving three theorems about problems in this class. These theorems are all of the form "given a data structure D for a decomposable searching problem we can transform D into a new data structure D' for a related problem". The correctness and complexity analysis of D are then used to establish the correctness and complexity of D'. We present transforms for converting a static structure into a dynamic structure, for adding "range variables" to queries, and for making preprocessing/query time tradeoffs. These transforms have already been used to develop a number of best known "theoretical" algorithms, and promise to be an important tool in software engineering.

1. Introduction

The young field of "Concrete Computational Complexity" has given birth to a large number of interesting and useful results in the past decade. These results have allowed us to apply the tools of mathematics to the problems of describing real-world costs of computations. Unfortunately, however, most of these results have been of a fairly limited scope; the literature is rich with theorems of the form "this problem is of this complexity" but has few statements such as "all problems in this class have this complexity". In this paper we define such a class of problems and prove three theorems about the complexity of problems in that class.

We investigate the problem of algorithms and data structures for searching and focus our attention on a class of searching problems we call *decomposable*. We give an algebraic definition of decomposable searching problems that allows one to test whether a given problem is in fact decomposable. We then give three constructions that allow one to transform a data structure for a "common" decomposable problem into a structure that solves a more "exotic" problem. We make no assumptions about the underlying "common" structure; we use only the fact that the problem is decomposable.

We see this work as one of the first steps towards Tarjan's [1978] goal of a "calculus of data structures". Although new results are presented, we see the main contributions of this paper more in its systematization of a broad class of results. We characterize the class by the *algebraic* structure of the problem statement, which is a fundamentally different kind of characterization than those used to identify other well-known complexity classes. The NP-Complete problems are defined by *reducibilities*, as are many diverse problems that are known to be equivalent in complexity to matrix multiplication. Lipton and Tarjan [1977] solve a diverse class of problems by giving a *method* applicable for many problems in a specific domain (planar graphs).

This paper describes the research on decomposable problems that was done up to the end of 1977. Since that time the author and James B. Saxe have found a number of further results in this area. This paper can therefore be viewed as an introduction; a more complete description of decomposable searching problems will appear in Bentley and Saxe [1978]. In Section 2 of this paper we present our model of searching and define the class of decomposable problems within that framework. The three main constructions are sketched in Sections 3, 4, and 5. In Section 6 we present conclusions.

2. Decomposable Problems

A *static* searching problem is usually given as follows: preprocess a set (or "file") F of N objects into a data structure D such that certain kinds of queries about F can be answered quickly. (In some contexts F is a multiset; we refer to it here as a set for brevity.) To analyze the structure D we give three functions of N: $S_D(N)$, the amount of storage required by D; $P_D(N)$, the preprocessing time required to build D; and $Q_D(N)$, the time required to answer a query. Unless explicitly stated otherwise, throughout this paper we will assume that we are measuring the "worst-case" complexity of these quantities.

The most well known example of a searching problem is usually called the *Member* problem, given as follows: preprocess N elements from a totally ordered set such that queries of the form "is x in set F?" can be answered quickly. A common solution to this problem is to store the elements of F in a vector sorted into increasing order and then answer queries by binary search. Analyzing this vector scheme shows that $P_V(N) = O(N \lg N)$, $S_V(N) = O(N)$, and $Q_V(N) = O(\lg N)$. A more difficult kind of searching problem is the *Nearest Neighbor* or *Post Office* problem which calls for preprocessing a set F of N points in the plane to facilitate queries of the form "what is the nearest neighbor in F to point x?" (where x is not necessarily in F). A data structure recently given by Lipton and Tarjan [1977] has $P_L(N) = O(N \lg N)$, $S_L(N) = O(N)$, and $Q_L(N) = O(\lg N)$.

Both Member and Nearest Neighbor are decomposable problems. Before formally describing the class of decomposable problems, we will illustrate certain features of the class with examples from these two problems. The first point to be made is that it is the Member and Nearest Neighbor *problems* themselves that are decomposable, and not the particular data structures or algorithms used to solve the problems. An informal definition of decomposability is that a search problem is decomposable if one can answer a query about set F by combining the answers to the query asked of sets A and B, where A and B are an (arbitrary) partition of F into two subsets. Member is decomposable because x is a member of F if and only if x is a member of A or x is a member of B, for any partitioning of F into A and B. Likewise Nearest Neighbor is decomposable because the distance from x to its nearest neighbor in F is the minimum of the distances from x to its nearest neighbors in A and B, once again for any partition. (We deal here with distance to nearest neighbor for tractability; extra bookkeeping will tell which point realizes that distance.)

We can now give a more formal definition of decomposability. We say that a searching problem is decomposable if the response to a query asking the relation of a new object x to set F can be written as

$$Q(x,F) = \mathop{\Box}_{f \in F} q(x,f).$$

We assume that $\Box$ is the repeated application of the binary operator $\square$ over all elements in its domain. For $\Box$ to be well defined mathematically $\square$ must be associative, commutative, and have an identity; for computational efficiency we require that $\square$ be computable in constant time. We can cast Member in this framework as

$$\mathrm{Member}(x,F) = \mathop{\mathrm{OR}}_{f \in F} \mathrm{equal}(x,f)$$

and Nearest Neighbor as

$$\mathrm{NN}(x,F) = \mathop{\mathrm{MIN}}_{f \in F} \mathrm{distance}(x,f).$$

It is clear that the queries that can be cast in this schema obey the informal definition of decomposability we gave above; for any partition of F into A and B, we know by the associativity and commutativity of $\square$ that

$$\begin{aligned} Q(x,F) &= \mathop{\Box}_{f \in F} q(x,f) \\ &= \square\ \Big(\mathop{\Box}_{g \in A} q(x,g),\ \mathop{\Box}_{h \in B} q(x,h)\Big) \\ &= \square\ (Q(x,A),Q(x,B))\,. \end{aligned}$$

More than twenty decomposable searching problems have been identified; we now mention a few of these. From problems on linearly ordered sets (where the file to be preprocessed contains elements from such a set) we have already seen the example of Member. Further examples are Successor (what is the least element in F greater than x?), Rank (how many elements in F are less than x?), and Count (how many elements in F have value x?). In Data Base problems, all of the problems that Rivest [1974] calls intersection queries are decomposable; these include secondary key retrieval, partial match searching, and range searching. Computational Geometry abounds with examples of decomposable problems. We have already mentioned Nearest Neighbor; related decomposable problems are Farthest Neighbor (which point is most distant from x?) and Fixed Radius Near Neighbors (which points are within some fixed distance d of x?). Queries dealing with more complicated geometric structures can also be decomposable; examples of such problems are given by Dobkin and Lipton [1976].

Although many searching problems are decomposable, some others are not. An example of a problem that is not decomposable is convex hull inclusion (preprocess N points in the plane; a query asks if a new point is within their convex hull). For any given point x within the convex hull of F it is not hard to find a partition of F into A and B such that x is not within the convex hull of either A or B. We can use this to prove that convex hull inclusion is not decomposable.

3. Dynamic Structures

In this section we turn our attention from static problems to dynamic problems. In static problems the data is organized once-and-for-all before any searches are done. In dynamic problems the set F (and the data structure D) are initially empty and elements are then added to F one-by-one. (The term dynamic sometimes implies that elements can also be deleted from F; we do not use it in that sense.) The most well known example of a dynamic data structure is the AVL tree described by Knuth [1973]. An AVL tree allows N elements to be inserted at a total cost of $O(N \lg N)$; at any point one can answer Member queries in $O(\lg N)$ time. In this section we will first develop a dynamic structure for the Nearest Neighbor problem, and then show how the techniques we employ can be used to convert any static structure for a decomposable problem into a dynamic structure.

There are two naive approaches to this problem. The first stores the points in the plane in a linked list, inserts a new point by appending it to the front of the list, and answers a query by examining every point in the list. Calling this structure B (for "brute force") we have

$$P_B(N) = S_B(N) = Q_B(N) = O(N).$$

A second naive solution calls for using the Lipton/Tarjan structure and rebuilding it after each insertion. This "rebuilding" scheme yields $P_R(N) = O(N^2 \lg N)$, $S_R(N) = O(N)$, and $Q_R(N) = O(\lg N)$. In choosing between these schemes one could achieve very good insertion time or very good query time, but not both. We will now develop a structure that has both.

Our faster structure will consist of a set of Lipton/Tarjan structures (which we abbreviate *LTS's*), each of a distinct size which is a power of two. Our structure is initially empty. When the first point is inserted, we build an LTS of size one. When the second point is inserted, we build the two points into an LTS of size two, discarding our previous LTS of size one. When the third point is inserted we have two LTSs (of sizes one and two), and after the fourth point is inserted we have one LTS of size four. Insertions proceed in a way analogous to binary counting: after the N-th element is inserted we have an LTS of size 2^j if and only if the j-th bit of the binary integer N is one. (This structure is similar to the binomial queues of Vuillemin [1978].) To answer a Nearest Neighbor query for a new point x we perform a nearest neighbor search for x in all the LTSs in our structure, then return the minimum of those as the answer to the query.

To analyze our structure (which we call L', because it is a transformation of the LTS L) we note that if L' contains N elements then we are using at most lg N LTSs. Since each of those is of size less than N we know that we can perform a nearest neighbor search on any of them in O(lg N) time. Therefore the time required to search is $Q_{L'}(N) = O(\lg^2 N)$. Since each of the LTSs require space linear in the number of elements it contains, the total space requirement of L' is $S_{L'}(N) = O(N)$. To count the total cost of having inserted N elements into L' is a bit more difficult. The total cost of building LTSs of size $m = 2^j$ is the cost of building one LTS of size m multiplied by the number of times the j-th bit in a binary word turns from zero to one in counting from one to N. If we assume that N is a power of two then a simple sum shows that $P_{L'}(N) = O(N \lg^2 N)$, and we can use this fact to show that $P_{L'}(N)$ is of the same order for N not a power of two.

The structure that we have used for dynamic Nearest Neighbor searching can be used to convert any static structure for a decomposable problem into a dynamic structure for that problem. We will assume that we are given a static structure D with performances $P_D(N)$, $S_D(N)$, and $Q_D(N)$. Our dynamic structure D' will consist of a set of static structures, each of distinct size a power of two, built as before in a manner analogous to binary counting. Analyzing the D' shows

$P_{D'}(N) = O[P_D(N) \lg N]$,
$Q_{D'}(N) = O[Q_D(N) \lg N]$, and
$S_{D'}(N) = O[S_D(N)]$.

The above analysis shows that we can convert a static structure to a dynamic structure with only a logarithmic increase in query and insertion times. We can achieve even better results for some structures. If the static query time $Q_D(N)$ grows faster than a polynomial in lg N, then we can show that $Q_{D'}(N)$ is bounded above by some constant times $Q_D(N)$. Likewise if $P_D(N)$ grows faster than N times a polynomial in lg N, then we will not incur the additional logarithmic factor in $P_{D'}(N)$. For certain structures we do not have to build the static structures over again from scratch--we can *merge* existing structures. In that case too we can avoid the extra logarithmic term in $P_{D'}(N)$. (We can give at least five examples of such structures.) We can also show that for some problems the increase of the logarithmic factor is not incurred if we measure average instead of worst-case query times.

This static-to-dynamic transformation has already been used to yield a number of new algorithms. Bentley, Detig, Guibas and Saxe [1978] describe "binomial lists", which were obtained by applying the transformation to sorted arrays. They show that with proper implementation the storage used by binomial lists is absolutely minimal and that their structure is optimal over the class of all structures using only minimal storage. In addition to theoretical interest, their structure is efficiently implemented and experiments show that it is competitive with other dynamic member structures in many applications. Another use of this transform has been given by Yao, Yao and Bentley [1978]. They reduced the complexity of calculating the "rank function" in a vector set from barely sub-quadratic to N times a polynomial in lg N by transforming a static maxima searching structure to dynamic.

Many other aspects of converting static search structures to dynamic will be described by Bentley and Saxe [1978]. In this section we have seen a transform that increases both query and processing costs by a factor of $O(\lg N)$. Bentley and Saxe display a whole set of transformations which add a factor of k to query time and a factor of $kN^{1/k}$ to processing time, for any fixed integer k. (For example, it is possible to convert a static structure to dynamic by adding a factor of 5 to query time and $5N^{1/5}$ to processing time; this might be useful if many queries were anticipated.) Further, they show that using only the properties of decomposability, these transformations are optimal to second-order terms. They then demonstrate "dual" transformations that add a factor of k to processing times at the cost of increasing query times by a factor of $(k^2/2)N^{1/k}$. They investigate the question of dynamic structures in which elements can be deleted as well as inserted and show that although deletion is provably infeasible in general, if $\square$ is invertible then deletion can be achieved at a cost of only a constant factor. They also show how the processing time between insertions can be bounded; this is useful in on-line systems. In related work, Bentley and Shaw [1978] have formally proved the correctness of the static-to-dynamic conversion discussed in this chapter by writing it as an Alphard form and proving its correctness in that context. Similar methods can be used to prove formally the correctness of all the transforms of Bentley and Saxe.

4. Adding Range Variables

In the last section we showed how to transform a static structure into a dynamic structure; in this section we will show how to transform a static structure for a particular query into a static structure for a related query. We will illustrate the construction by first applying it to the planar Nearest Neighbor problem, and then considering the general case. The related searching problem that we will solve is most easily stated if the points in the plane are viewed as cities, each of which has an associated population. We are to preprocess the N cities and their populations. A query will give a point x in the plane and a population *range* (described by upper and lower bounds), and the search must determine which of the cities in the desired population range is the closest to point x[1]. (So a query might ask "of all the cities with population between 60,000 and 120,000, which is the closest to point x?".)

The data structure we use to solve this problem is a tree that has LTSs in all of its nodes. The root of our tree contains an LTS representing all the cities in F. The left son of the root will represent all cities with population less than the median population, and the right son will represent the cities with populations greater than the median. In each of those nodes there will be an LTS representing half of the cities. This partitioning continues so that on the i-th level of the tree there are 2^i LTSs, each representing cities contiguous in the population dimension. To answer a query for the nearest neighbor to point x in population range R we will search for the nearest neighbor to x in a subset of the LTSs (the union of which is all of the cities in the range), and then return the minimum of the reported distances. Specifically we search all the the LTSs that represent a population range contained in R and having fathers with range not containing R. We can easily describe exactly which LTSs will be searched by a recursive algorithm which visits the nodes in the tree containing the relevant LTSs.

---

[1] This search is of the same form as one described by Knuth [1973, p. 550] as "more complicated than those already quite difficult and therefore usually not considered".

The first step in analyzing our structure is to note that the tree we build is lg n levels deep. The time required for building the LTSs on each level is bounded above by O(N lg N), so we have $P_{L'}(N) = O(N \lg^2 N)$ (L' now represents the LTS with range restriction capability added). Likewise the storage required on any level is O(N), so $S_{L'}(N) = O(N \lg N)$. Since at most two LTSs are searched on any level (at cost bounded above by O(lg N)), we have $Q_{L'}(N) = O(\lg^2 N)$. Thus we see that our new structure adds a factor of lg N to each of the cost functions we measure.

We consider now the general case in which each element f of the set F contains an additional population dimension written p(f). The modified query we want to answer (for object x and range R) is

$$Q'(x,R,F) = \mathop{\square}_{\substack{f\in F\\ p(f)\in R}} q(x,f).$$

The structure we use is the tree structure described above in which each node of the tree contains one of the original structures D. Analyzing the resulting structure D' as before shows that a factor of lg N is added to each of the cost functions giving

$$P_{D'}(N) = O[P_D(N) \lg N],$$
$$S_{D'}(N) = O[S_D(N) \lg N], \text{ and}$$
$$Q_{D'}(N) = O[Q_D(N) \lg N].$$

This transformation can be used to solve the "range searching" problem defined by Knuth [1973, p. 554]. In this problem we preprocess N points in k-space and answer subsequent queries asking for all points that have every key value in some specified range (that is, for all points lying in some rectilinearly-oriented hyperrectangle). Range searching in one dimension can be accomplished by use of a sorted vector. We can use this structure as a "base" and add k-1 additional range variables by the transform of this section. We then apply the "speedup" of merging structures to yield the "range tree" structure R with performances

$$P_R(N,k) = O(N (\lg N)^{k-1}),$$
$$S_R(N,k) = O(N (\lg N)^{k-1}), \text{ and}$$
$$Q_R(N,k) = O((\lg N)^k + F)$$

where F is the number of points found in response to the query. This structure is the best known structure for range searching, and was discovered through the use of decomposable problems. This transformation can also be used to derive the data structures for "Maxima Searching" and "ECDF Searching" described by Bentley and Shamos [1977].

Further aspects of adding range variables will be described by Bentley and Saxe

[1978]. Just as in the static-to-dynamic transformation there were a set of conversions, so there are in this transformation also. We have seen in this section a transformation which increases the preprocessing, storage, and query costs by a factor of O(lg N). Bentley and Saxe show a set of transformations that increase preprocessing and storage costs by a factor of k and query costs by a factor of $O(N^{1/k})$. The dual transform adds a factor of k to query costs and increases preprocessing and storage requrements by $O(N^{1/k})$. They also show that these increases are not always incurred. They can be avoided in exactly the same way as in the static-to-dynamic conversion: by merging underlying structures, if the underlying functions are fast growing, and if average query times are considered. The transformations that we have described in this section add a range variable to a static structure, yielding a new static structure. Lueker [1978] has described a new transformation that is applied to a dynamic sructure and yields a new dynamic structure in which range variables can be specified. He has used this transform to produce best-known structures for dynamic range searching, ECDF searching, and maxima searching.

5. Preprocessing/Query Time Tradeoffs

In this section we examine the final construction for decomposable problems. Unlike the two other constructions, this construction is not due to the author. It has been used previously in a number of algorithms, and it is included here because its scope of applicability is precisely the decomposable problems. The construction is appropriate for a structure that has very high preprocessing time or storage; it allows us to develop a class of structures with decreased preprocessing and storage requirements at the cost of increased search time.

The data structure makes use of *clusters*. The preprocessing partitions the N elements of F (at random) into N/c clusters of c elements each, then applies the preprocessing algorithm of D to each of the clusters to build N/c structures. To answer a query we search each of the clusters and then combine those answers (by decomposability) to form an answer to the original query. Analyzing the resulting structure D' shows

$Q_{D'}(N) = (N/c)\ Q_D(c)$,
$S_{D'}(N) = (N/c)\ S_D(c)$, and
$P_{D'}(N) = (N/c)\ P_D(c)$.

Notice that a continuum of performances is achievable.

This transform has been used by A. Yao [1977] to achieve a number of new results. He used as "underlying" structures those presented by Dobkin and Lipton [1976], that had logarithmic search times but preprocessing and storage that were (a large) polynomial in N. By choosing c appropriately (as a function of N) Yao was able to achieve the first sub-quadratic algorithms for many multidimensional problems. Yao's methods can be applied to many other problems which do not *prima facie* appear to be searching problems. These are problems which ask a "decomposable function" to be computed over every element in a set, and this can be reduced to performing N decomposable searches.

Yao's methods can then be applied to yield sub-quadratic algorithms. There are other applications of this transformation of structures in more typical searching contexts. In many applications the available storage is bounded, so we can use these methods to design a structure that will use exactly as much storage as is available. If we know in a certain application exactly how many searches will be made, then we can choose the cluster size to minimize the total cost of preprocessing and searching.

6. Conclusions

To summarize this paper we have seen at least two types of results. On a higher level we have seen three transforms that can be applied to searching structures for decomposable problems: converting a static structure to dynamic, adding "range variables" to a structure, and making preprocessing/query time tradeoffs. These transforms have already been used to yield results on a more concrete level. These include the currently best known "theoretical" algorithms for such problems as the rank function, member searching, range searching, and minimal spanning trees. These transforms can also be used as software engineering tools: each transform is is easily coded and would be a valuable entry in a "software engineering handbook".

Throughout this paper we have mentioned other work on decomposable problems that has already been done and will be reported by Bentley and Saxe [1978]. Much further work, however, remains to be done. Open problems include identifying additional decomposable problems, showing more efficient transforms than those we discussed, giving new types of transforms, showing the optmality of transforms, exact (Knuthian) analysis of transforms, and the development of software implementing these transforms. Perhaps the most obvious open problem, though, is to identify other classes of problems and prove general results concerning problems in the class.

In conclusion, this paper contains three kinds of results. The first kind of result that we have seen is a set of particular algorithms; the framework of decomposable problems has been useful in both the discovery and presentation of these algorithms. The second kind of result we have seen are the three particular transforms, and the different tradeoffs available within each transform. The final kind of result in this paper is the definition of the class of decomposable problems. To the author's knowledge, this is the first example of a class of problems that is defined algebraically and for which a general set of conversions is known. This kind of result promises to be valuable both in proving theorems about the asymptotic complexity of computational problems and in making software engineering more of an engineering discipline.

Acknowledgements

The author gratefully acknowledges helpful conversations with David Jefferson and Andrew and Frances Yao. The contributions of James Saxe to the author's understanding of decomposable problems (and thereby to the presentation of this introductory paper)

are too numerous to mention; a simple "thanks" will have to suffice.

References

Bentley, J. L., D. Detig, L. Guibas, and J. B. Saxe [1978]. An optimal data structure for minimal-storage dynamic member searching, in preparation.

Bentley, J. L. and J. B. Saxe. Decomposable searching problems, in preparation.

Bentley, J. L. and M. Shaw [1978]. A class of correct and efficient transformations for converting data structures from static to dynamic, in preparation.

Bentley, J. L. and M. I. Shamos [1977]. "A problem in multivariate statistics: algorithm, data structure, and applications", *Proceedings of the Fifteenth Allerton Conference on Communication, Control and Computing*, pp. 193-201.

Dobkin, D. and R. J. Lipton [1976]. "Multidimensional searching problems", *SIAM Journal of Computing* 5, pp. 181-186.

Knuth, D. E. [1973]. *The Art of Computer Programming, volume 3: Sorting and Searching*, Addison-Wesley, Reading, Mass.

Lipton, R. J. and R. E. Tarjan [1977]. "Applications of a planar separator theorem", *Eighteenth Symposium on the Foundations of Computer Science*, pp. 162-170.

Lueker, G. S. [1978]. A transformation for adding range restriction capability to dynamic data structures for decomposable searching problems, UC Irvine Technical Report.

Rivest, R. L. [1974]. Analysis of associative retrieval algorithms, Stanford Computer Science Department Report STAN-CS-74-415, 102 pp.

Tarjan, R. E. [1978]. "Complexity of combinatorial algorithms", *SIAM Review 20*, 3, pp. 457-491 (July 1978).

Vuillemin, J. [1978]. "A data structure for manipulating priority queues," *Comm. of the ACM, 21*, 4, pp. 309-315 (April 1978).

Yao, A. C. [1977]. "Fast algorithms for finding minimum spanning trees in k dimensions", *Proceedings of the Fifteenth Allerton Conference on Communication, Control and Computing*, pp. 553-556.

Yao, A. C., F. F. Yao, and J. L. Bentley [1978]. On computing the rank function of a set of vectors, to appear.

A LOWER WORST-CASE COMPLEXITY FOR SEARCHING A DICTIONARY

D.S. HIRSCHBERG
Rice University
Electrical Engineering Department
Houston, Texas

ABSTRACT

It is shown that $k(p+3)/2 + p-2$ letter comparisons suffice to determine whether a word is a member of a lexicographically ordered dictionary containing $2^p - 1$ words of length k. This offers a potential savings (compared to worst case complexity of binary search) that asymptotically approaches 50 percent.

We consider the problem of searching a dictionary containing n words, each word consisting of at most k letters. It is assumed that the alphabet is ordered and that the dictionary is in lexicographic order.

Two letters can be compared (to determine which has lower rank in the alphabet) in unit time. Our goal is to exhibit an algorithm for searching the dictionary that has minimal worst case time complexity.

We denote the ordered set of words (k-vectors) by $\overline{B}$ and refer to the i-th word (vector) by B_i. We wish to determine if the vector A is in $\overline{B}$. Assume that $n = 2^p - 1$, i.e. $p = \lceil \log(n+1) \rceil$.

Let B_m be the "middle" vector ($m = 2^{p-1}$), let B_q be the "quarter" vector ($q = 2^{p-1} + 2^{p-2}$).

Using binary search, i.e. comparing A first with B_m and then, depending upon the result, comparing A with either B_q or with B_r etc. will lead to a worst case complexity of kp comparisons.

In a recent paper [2], Hirschberg has exhibited an algorithm, VECTOR-SEARCH, that has a worst case complexity of less than $k(p+2-\log k)$. For many values of n and k, this algorithm offers a potential savings of from 5 to 25 percent.

Research supported by NSF grant MCS-76-3933. Author's current address: Department of Electrical Engineering, Rice University, Houston, Texas, 77001.

Let there have been x "equal" comparisons between A and B_0 (if no comparisons were made between A and B_0 then x = 0) and y "equal" comparisons between A and B_{n+1}. Without loss of generality, assume that $x \leq y$ (the case $x > y$ will lead to symmetric arguments).

For the algorithm we shall next consider, which we call modified binary search, let T(x,y,p) be the minimum number of comparisons required for x, y, and p as defined above and for k (the length of the vectors) implicit. We shall demonstrate a savings that asymptotically approaches 50 percent of the worst case complexity as compared to the binary search method.

Note that for all $1 \leq i \leq n$ we need not compare $a_j : b_{ij}$ for $j \leq x$ since the result must be "equal" from the results of previous comparisons and the fact that $\overline{B}$ is in lexicographic order.

Our algorithm first compares components of A with components of B_q (starting at component x + 1) until it is determined whether $A < B_q$, $A = B_q$, or $A > B_q$.

Case 1. If $A < B_q$ because $a_{t+1} < b_{q,t+1}$ $(x \leq t < k)$ then

$$T(x,y,p) = T(x,t,p-2) + t+1-x \quad (1)$$

Case 2. If $A > B_q$ because $a_{t+1} > b_{q,t+1}$ then compare A with another vector depending on the value of t.

Case 2a. If $t \leq y$ then compare A with B_m (starting with component t + 1)

2aa. If $A < B_m$ because $a_{h+1} < b_{m,h+1}$ $(x \leq t \leq h < k)$ then

$$T(x,y,p) = T(t,h,p-2) + h+2-x \quad (2)$$

2ab. If $A > B_m$ because $a_{h+1} > b_{m,h+1}$ $(x \leq t \leq h < k)$ then

$$T(x,y,p) = T(\min\{h,y\}, \max\{h,y\}, p-1) + h+2-x \quad (3)$$

2ac. If $A = B_m$ then

$$T(x,y,p) = k+1-x \quad (4)$$

Case 2b. If $t \geq y$ then compare A with B_r (starting with component y+1)

2ba. If $A < B_r$ because $a_{h+1} < b_{r,h+1}$ $(y \leq h,\ t < k)$ then

$$T(x,y,p) = T(\min\{h,t\}, \max\{h,t\}, p-1) + t+1-x + h+1-y \quad (5)$$

2bb. If $A > B_r$ because $a_{h+1} > b_{r,h+1}$ $(y \leq h,\ t < k)$ then

$$T(x,y,p) = T(y,h,p-2) + t+1-x + h+1-y \quad (6)$$

2bc. If $A = B_r$ then

$$T(x,y,p) = t+1-x + k-y \quad (7)$$

Case 3. If $A = B_q$ then

$$T(x,y,p) = k-x \quad (8)$$

The resulting problem has either half or quarter the range (in $\overline{B}$) of the original problem. If the range is reduced to zero then terminate with the solution that A is not in $\overline{B}$. If the range consists of one vector B then compare $a_{x+1} \ldots a_k$ with the corresponding $k-x$ components of B. Otherwise, recursively apply the modified binary search algorithm.

Theorem. For $p \geq 1$, $T(x,y,p) \leq k(p+3)/2 + p-2 - x - y$

Proof. By induction on p. Note that $T(x,y,0) = 0$. The theorem, for $p = 1$, is true since $T(x,y,1) = k-x$ (the algorithm treats this case separately). The theorem, for $p = 2$, is also true as can be seen by substituting for $T(\cdot,\cdot,0)$ and $T(\cdot,\cdot,1)$ in the righthand side of the formulas given in all of the cases, one of which must occur.

For the inductive step, $T(x,y,p)$ is bounded from above by the maximum of the cases. In all cases, substituting in the assumed complexity for T (with smaller value of the third argument) substantiates that the theorem holds.

The table below illustrates typical percentage savings (worst case analysis) of VECTOR-SEARCH and modified binary search as compared to binary search.

n (p)	k	Binary search	VECTOR-SEARCH (%)	modified bin. search (%)
125 (7)	4	28	26 (7.1)	25 (10.7)
	8	56	46 (17.9)	45 (19.6)
	16	112	78 (30.4)	81 (27.7)
1000 (10)	4	40	38 (5.0)	34 (15.0)
	8	80	70 (12.5)	60 (25.0)
	16	160	126 (21.2)	112 (30.0)
10^6 (20)	4	80	78 (2.5)	64 (20.0)
	8	160	150 (6.2)	110 (31.2)
	16	320	286 (10.6)	202 (36.9)

We have shown that, approximately, $k(\log_2 n)/2$ letter comparisons suffice to search a dictionary. Research currently in progress indicates that this upper bound on the complexity of dictionary searching may be reduced to $k(\log_2 n)/\log_2 k$. A lower bound of $k + \log_2 n$ can easily be shown (see [2]) and it is an open problem to narrow the gap between these bounds.

REFERENCES

1. D. Dobkin and R.J. Lipton. Multidimensional Searching Problems. *SIAM J. Computing* 5:2 (June 1976), pp. 181-186.
2. D.S. Hirschberg. On the Complexity of Vector Searching. Rice University Department of Electrical Engineering Technical Report #7807, June 1978.
3. D.E. Knuth. *The Art of Computer Programming, Vol. 3*. Addison-Wesley, 1973, pp. 200-204.
4. V.V. Raghavan and C.T. Yu. A Note on a Multidimensional Searching Problem. IPL 6:4 (August 1977), pp. 133-135.

Inferring a Tree from Lowest Common Ancestors With an Application to the Optimization of Relational Expressions

A. V. Aho
Bell Laboratories
Murray Hill, New Jersey

and

Y. Sagiv*, T. G. Szymanski†, and J. D. Ullman‡
Princeton University
Princeton, New Jersey

We present an algorithm for constructing a tree to satisfy a set of lineage constraints on common ancestors. We then apply this algorithm to synthesize a relational algebra expression from a simple tableau, a problem arising from the theory of relational databases.

1. A Tree Discovery Problem

In a rooted tree, the lowest common ancestor of two nodes x and y, denoted (x, y), is the node a that is an ancestor of both x and y such that no proper descendant of a is also an ancestor of both x and y. Suppose we are told a tree T has leaves numbered $1, 2, \ldots, n$, and we are given a set of constraints on the lowest common ancestors of certain pairs of leaves. In particular, suppose the constraints are of the form $(i, j) < (k, l)$ where $i \neq j$ and $k \neq l$, meaning that the lowest common ancestor of i and j is a proper descendant of the lowest common ancestor of k and l. Note that the order of i and j in (i, j) and of k and l in (k, l) are irrelevant. From a set of constraints of this form, can we reconstruct T, or determine that no such tree exists?

Example 1. Suppose we are given the constraints

$$\begin{aligned}(1, 2) &< (1, 3)\\ (3, 4) &< (1, 5) \qquad (1.1)\\ (3, 5) &< (2, 4)\end{aligned}$$

One possible tree T consonant with these constraints is

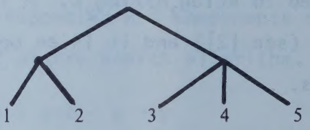


Fig. 1. A solution to constraints (1.1).

However, if we add the constraint $(4, 5) < (1, 2)$, then it can be shown that no tree can simultaneously satisfy all these constraints. □

In this paper we give an efficient algorithm for solving this problem. We then present an application of this algorithm to the problem of synthesizing a relational algebra expression from a simple tableau.

2. Finding the Tree

The central idea behind the solution is to determine for a potential tree T the sets of leaves that are descendants of each child of the root of T. Call these sets $S_1, S_2, \ldots, S_r$. We may assume $r \geqslant 2$, since if a tree satisfying a set of constraints exists, we can find another tree satisfying the same set of constraints if we merge each node having one child with its child. In Example 1, we have $S_1 = \{1, 2\}$ and $S_2 = \{3, 4, 5\}$.

There are two conditions that these sets must satisfy for each constraint $(i, j) < (k, l)$.

(1) i and j must be in the same set. Otherwise (i, j) is the root of T, and the root cannot be a proper descendant of (k, l).

---

* Present address: University of Illinois, Urbana, Illinois. Work supported in part by NSF grant MCS-76-15255 and a grant from Bell Laboratories.

† Present address: Bell Laboratories, Murray Hill, New Jersey. Work supported in part by NSF grant DCR-74-21939.

‡ Work supported in part by NSF grant MCS-76-15255.

```
procedure BUILD($S$, $C$);
if $S$ consists of a single node $i$ then
        return the tree consisting of node $i$ alone
else
    begin
        compute $\pi_C = S_1, S_2, \ldots, S_r$;
        if $r = 1$ then
                return the null tree
        else
            for $m := 1$ to $r$ do
                begin
                    $C_m := \{ (i, j) < (k, l) \text{ in } C \mid i, j, k, l \text{ are in } S_m \}$;
                    $T_m :=$ BUILD($S_m$, $C_m$);
                    if $T_m$ = the null tree then
                        return the null tree
                end;
        /* if we reach here a tree exists */
        let $T$ be the tree with a new node for its root and
            whose children are the roots of $T_m$, $1 \leqslant m \leqslant r$;
        return $T$
    end
end BUILD
```

Fig. 2. The procedure BUILD.

(2) Either k and l are in different sets, or i, j, k and l are all together in one set. Otherwise, (i, j) cannot be a proper descendant of (k, l).

We shall also show that conditions (1) and (2) are sufficient. Thus, if we can partition the nodes into two or more sets satisfying (1) and (2), and if we can recursively build trees for each set, then a tree exists; otherwise, one does not.

Given a set of constraints C, we define a partition π_C on the leaves $1, 2, \ldots, n$ using the following rules:

(1) If $(i, j) < (k, l)$ is a constraint, then i and j are in one block of π_C.

(2) If $(i, j) < (k, l)$ is a constraint, i and j are in one block, and k and l are in one block, then i, j, k and l are all in the same block.

(3) No two leaves are in the same block of π_C unless it follows from (1) and (2).

Example 2. π_C for the constraints given in Example 1 is $\{1, 2\}$, $\{3, 4, 5\}$. Note that no instance of rule (2) is used. If we add to (1.1) the constraint $(4, 5) < (1, 2)$, to obtain the set

$$\begin{array}{l}(1, 2) < (1, 3) \\ (3, 4) < (1, 5) \\ (3, 5) < (2, 4) \\ (4, 5) < (1, 2)\end{array}$$

then by rule (2) the two blocks must be merged, yielding a trivial partition. Since a necessary condition for the existence of a tree is that π_C have more than one block, we can immediately infer that this new set of constraints is not satisfied by any tree. □

In Fig. 2 we present a recursive algorithm to build a tree T satisfying a set of constraints C on a nonempty set of nodes S. It returns the null tree if no tree exists. The basic idea is to compute the partition π_C, check that it has at least two blocks $S_1, S_2, \ldots, S_r$, $r \geqslant 2$, and construct the sets of constraints C_m, $1 \leqslant m \leqslant r$, such that C_m is C restricted to those constraints $(i, j) < (k, l)$ in C for which i, j, k and l are all in S_m.

Theorem 1. If BUILD(S, C) returns a non-null tree T, then T satisfies the constraint set C.

Proof. We proceed by induction on the size of S. The basis, one node, is trivial. Suppose it is true for all sets smaller than S. Consider an arbitrary constraint $(i, j) < (k, l)$ in S. By rule (1) in the definition of π_C, i and j are in the same block, and therefore (i, j) is not the root. Two cases arise depending on whether k and l are in the same or in different blocks.

Case 1. If k and l are in the same block, then all of i, j, k and l are in some set S_m, by rule (2) in the definition of π_C. Also, $(i, j) < (k, l)$ is in C_m. By the inductive hypothesis, BUILD(S_m, C_m) produces a tree that satisfies $(i, j) < (k, l)$, so the final tree satisfies that constraint also.

Case 2. If k and l are in different blocks, then (k, l) is the root of the tree, and (i, j), we already deduced, is not the root, so $(i, j) < (k, l)$ is surely satisfied. □

Lemma 1. Let T be any tree satisfying constraint set C. Then each block of π_C is wholly contained within the subtree rooted at some child of the root of T.

Proof. Let us consider any fixed order in which we may apply rules (1) and (2) to construct π_C, starting with each leaf in a block by itself. We shall show, by induction on the number of applications of the rules, that each block of the partition being formed always lies wholly within the subtree of some child of the root of T.

Basis. Zero applications. Each leaf is in a block by itself, so the result is trivial.

Induction. Case 1. Rule (1) is applied to i and j, merging their blocks, say S_1 and S_2. Then S_1 and S_2 are each contained in the subtrees rooted at n_1 and n_2, two children of the root. If $n_1 = n_2$, we are done. If $n_1 \neq n_2$, then (i, j) is the root of T. But then, no constraint of the form $(i, j) < (k, l)$ can be satisfied by T, violating our assumption that T satisfies C.

Case 2. Rule (2) is applied to constraint $(i, j) < (k, l)$, where i and j are in some block S_1, and k and l are in some block S_2 different from S_1. Let n_1 and n_2 be the children of the root whose subtrees contain S_1 and S_2, respectively. If $n_1 \neq n_2$, then (i,j) is a descendant of n_1 and (k, l) is a descendant of n_2, violating the assumption that T satisfies $(i, j) < (k, l)$. □

Theorem 2. If there is a tree satisfying the constraint set C, then BUILD(S, C) returns some non-null tree.

Proof. Suppose not, and assume that T with the constraint set C is as small a counterexample as there is. Suppose BUILD(S, C) returns the null tree. Then either π_C has only one block, or BUILD(S_m, C_m) returns the null tree for some m. In the first case, it follows from Lemma 1 that the root of T has only one child. But then T', which is T with the root removed, provides a smaller counterexample to the theorem.

In the second case, where BUILD(S_m, C_m) returns the null tree, let S_m be contained in the subtree rooted at child n of the root of T. Define T'' to be the subtree of T with root n, after deleting all leaves not in S_m and any interior nodes none of whose descendant leaves are in S_m. Then T'' satisfies C_m, T'' is smaller than T, yet BUILD(S_m, C_m) does not construct a non-null tree. Thus T together with C was not the smallest counterexample to the theorem, as supposed.

As a single node cannot be a counterexample to the theorem, we conclude that the theorem has no smallest counterexample. Since a counterexample, if one exists, is finite, there can be no counterexample at all, from which we conclude the theorem. □

Example 3. Suppose we are given the constraints

$(1,3) < (2,5)$
$(1,4) < (3,7)$
$(2,6) < (4,8)$
$(3,4) < (2,6)$
$(4,5) < (1,9)$
$(7,8) < (2,10)$
$(7,8) < (7,10)$
$(8,10) < (5,9)$

Rule (1) produces partition $\{1,3,4,5\}$, $\{2,6\}$, $\{7,8,10\}$, $\{9\}$. By rule (2), we must merge the first two blocks because of constraint $(3,4) < (2,6)$. Thus the tree constructed top-down begins as in Fig. 3(a). The constraints germane to $\{1, 2, \ldots, 6\}$ are $(1,3) < (2,5)$ and $(3,4) < (2,6)$, while only $(7,8) < (7,10)$ is germane to $\{7, 8, 10\}$. The partition for the former is $\{1, 3, 4\}$, $\{2\}$, $\{5\}$, $\{6\}$, and for the latter $\{7, 8\}$, $\{10\}$. The second level of tree construction is shown in Fig. 3(b). No constraints are applicable to any of the remaining blocks, so at the next level, each block of more than one leaf is partitioned into singletons. The final tree is shown in Fig. 3(c). □

3. Complexity of the Tree Synthesis Algorithm

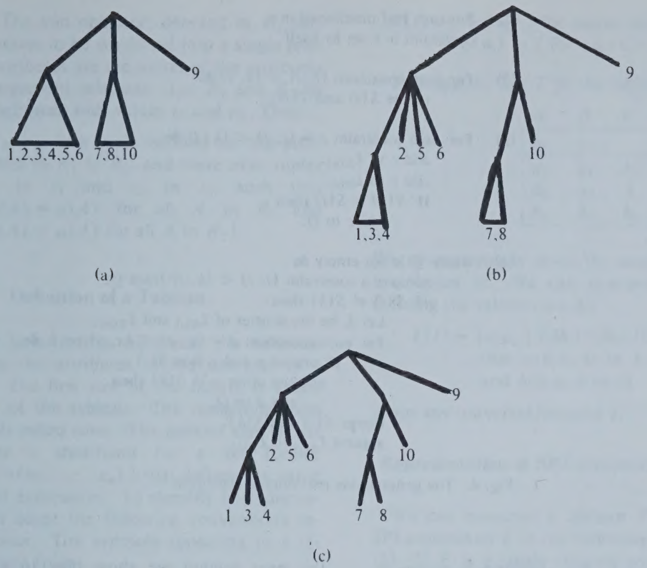
The running time of the tree synthesis algorithm of Section 2 is highly dependent on the method used to partition the set of constraints. In this section, we shall first analyze the running time of BUILD as a function of n, the number of constraints, and f, a function specifying the time needed to partition a set of constraints. We shall see that the use of the naive partitioning algorithm leads to an $O(n^3)$ implementation of BUILD. By using a more sophisticated partitioning algorithm, the overall running time can be reduced to $O(n^2 \log n)$. Moreover, by imposing the restriction that all constraints be of the form $(i, j) < (i, k)$, we can further reduce the running time to $O(n^2)$. Section 4 of this paper presents a database application in which problems of this restricted form arise naturally.

Lemma 2. Let $f(n)$ be the time needed to partition a set of n constraints subject to rules (1) and (2) of the previous section. If f is monotonically nondecreasing, then the time consumed by BUILD when applied to a set of n constraints is $O(nf(n))$.

Proof. The worst-case running time occurs when the algorithm succeeds in synthesizing a non-null tree, so we shall restrict our attention to analyzing this case. We shall proceed by assigning a cost to each node of the tree and summing the costs.

Since the top level call on BUILD involves n constraints, each of which can introduce at most $4n$ leaves into the tree, the synthesized tree has at most $4n$ leaves. (Leaves not mentioned in any constraint can be made children of the root of the final tree.) Because the branching factor at each internal node is at least 2, the number of internal nodes cannot exceed $4n-1$.

At each internal node, BUILD partitions some subset of the original constraints and then spends a constant amount of time (exclusive of recursive calls) processing each block of the partition. Since f is monotonic, $f(n)$ is certainly an upper bound on the cost of performing each partition. Moreover, since f must grow at least linearly with its argument, we can neglect the cost of distributing the constraints over the blocks of the partition when compared with $f(n)$ and charge each internal node $O(f(n))$ units of time. Thus the time charged to all internal nodes is $O((4n-1)f(n)) = O(nf(n))$. Since BUILD spends a constant amount of time at



(a) (b) (c)

Fig. 3. Construction of a tree.

each leaf, the contribution of the leaves to the running time is $O(n)$ for a total over all nodes of $O(nf(n))$. □

The problem of producing a fast implementation of BUILD thus reduces to one of producing a fast partitioning algorithm. Let us first first address this problem for the special case mentioned earlier.

Theorem 3. If all constraints are of the form $(i, j) < (i, k)$, then BUILD can be implemented to run in $O(n^2)$ time, where n is the number of constraints.

Proof. By Lemma 2, it suffices to show how to partition a set of n constraints in $O(n)$ time. When all constraints are of the form $(i, j) < (i, k)$, rule (2) of the partition constructing algorithm of Section 2 need never be applied explicitly. To see this, suppose that i and k are in the same block. By rule (1), i and j must also be in the same block, and thus by transitivity, i, j, and k are all in the same block. Accordingly, it suffices to implement rule (1) alone.

If we take each leaf appearing in one or more constraints to be a node, and each constraint $(i, j) < (i, k)$ to represent an edge between i and j, then we shall have a graph (actually, a multigraph) whose connected components represent the blocks of the partition we are looking for. Since it is easy to find connected components in time proportional to the number of edges [1], and the graph described above has at most n edges, we conclude that partitioning can be done in $O(n)$ time. □

Theorem 4. If no restrictions are placed on the form of the constraints, then BUILD can be implemented to run in $O(n^2 \log n)$ time.

Proof. By Lemma 2, it suffices to show how to partition a set of n constraints in $O(n \log n)$ time. Blocks of the partition will be maintained using a set merging algorithm that supports the following operations:

(1) Given an element i, *find* the set $S(i)$ that i currently belongs to.

(2) Given two sets $S(i)$ and $S(j)$, *merge* them together and assign them a common name.

One of the set merging algorithms described in [6] permits a find operation to be performed in constant time and a merge operation to be performed in time proportional to the size of the smaller of the two sets being merged. Thus, if we start with a collection of singleton sets, a sequence of n merges can be performed in time $O(n \log n)$.

Associated with each block B of the partition will be a list L_B consisting of all constraints $(i, j) < (k, l)$ for which either k or l is currently in B. We assume that the length of each list is stored with it and is available in constant time. We shall also employ a queue Q of constraints that

```
(1) Put each leaf mentioned in a
    constraint in a set by itself.

(2) for each constraint $(i, j) < (k, l)$ do
        merge $S(i)$ and $S(j)$;

(3) for each constraint $c = (i, j) < (k, l)$ do
        add $c$ to $L_{S(k)}$;
        add $c$ to $L_{S(l)}$;
        if $S(k) = S(l)$ then
            add $c$ to $Q$;

(4) while $Q$ is not empty do
        remove a constraint $(i, j) < (k, l)$ from $Q$;
        if $S(i) \neq S(k)$ then
            Let $L$ be the shorter of $L_{S(i)}$ and $L_{S(k)}$;
            for each constraint $d = (u, v) < (x, y)$ on $L$ do
                if one of $x$ and $y$ is in $S(i)$
                  and the other is in $S(k)$ then
                    add $d$ to $Q$;
            merge $S(i)$ and $S(k)$;
            append $L_{S(i)}$ to $L_{S(k)}$;
```

Fig. 4. The general case partitioning algorithm.

remain to be processed. The algorithm is shown in Fig. 4.

Before proceeding, notice that a constraint $(u, v) < (x, y)$ can only be placed on the queue once. This can occur either in Step (3) or in Step (4) just before $S(x)$ is merged with $S(y)$.

The time spent in Step (1) is clearly $O(n)$. Step (2) takes time $O(n\log n)$ because it involves at most n merge operations. Step (3) requires $O(n)$ time. The time expended by the the inner **for** loop of Step (4) is $O(n\log n)$ because each copy of a constraint is only considered in this loop when it is being moved from a shorter to a longer list. Since each constraint can move at most $\log_2 2n$ times and there are exactly 2 copies of each constraint, the amount of work done in the inner **for** loop is $O(n\log n)$. The time spent in Step (4), exclusive of the time spent in the inner **for** loop, is $O(n\log n)$ because no copy of a constraint is ever enqueued more than once. Thus the total time spent in Step (4) is $O(n\log n)$. □

4. Relational Algebra Expressions and Tableaux

In [5], Codd introduced relational algebra as a notation for expressing database queries. In [2, 3] a class of relational expressions called SPJ-expressions was studied in which the operands of an expression represent relations and the operators are the relational algebra operations select, project and natural join. In [3] it was shown that the value of an SPJ-expression could be represented in terms of a two-dimensional matrix called a *tableau*. Left open were the questions of how to construct an expression from a tableau and how to determine whether a tableau comes from an expression. In the next section we show how to answer these questions for the important special case of simple tableaux.

We first introduce the relevant definitions. We assume a database consisting of a set of two-dimensional tables called *relations*. The columns of a table are labeled by distinct *attributes* and the entries in each column are drawn from a fixed *domain* for that column's attribute. The ordering of the columns of a table is unimportant. Each row of a table is a mapping from the table's attributes to their respective domains. A row is often called a *tuple* or *record*. If r is a relation that is defined on a set of attributes that includes A, and if μ is a tuple of r, then $\mu(A)$ is the value of the A-component of μ.

A *relation scheme* is the set of attributes labeling the columns of a table. When there is no ambiguity, we shall use the relation scheme itself as the name of the table. A relation is just the "current value" of a relation scheme. The relation is said to be *defined on* the set of attributes of the relation scheme. The operators select, project and join are defined as follows.

(1) *Select.* Let r be a relation on a set of attributes X, A an attribute in X, and c a value from the domain of A. Then the *selection* $A = c$, written $\sigma_{A=c}(r)$, is the subset of r having value c for attribute A.

(2) *Project.* Let r be a relation on a set of attributes X. Let Y be a subset of X. We define $\pi_Y(r)$, the *projection* of r onto Y, to be the relation obtained by removing all the components of the tuples of r that do not belong to Y and removing duplicate tuples.

(3) *Join.* The join operator, denoted by $\bowtie$, permits two relations to be combined into a single relation whose attributes are the union of the attributes of the two argument relations. Let R_1 and R_2 be two relation schemes with values r_1 and r_2. Then

$$r_1 \bowtie r_2 = \{\mu \mid \mu \text{ is a tuple defined on the attributes in } R_1 \cup R_2 \text{, and there exist tuples } \nu_1 \text{ in } r_1 \text{ and } \nu_2 \text{ in } r_2 \text{, such that } \nu_1(A) = \mu(A) \text{ for all } A \text{ in } R_1 \text{ and } \nu_2(A) = \mu(A) \text{ for all } A \text{ in } R_2\}$$

Definition of a Tableau

A tableau is a matrix in which the columns correspond to the attributes of the universe in a fixed order. The first row of the matrix is called the *summary* of the tableau. The remaining rows are exclusively called *rows*. The general idea is that a tableau is a shorthand for a set former $\{ a_1 \cdots a_n \mid \psi(a_1, \ldots, a_n) \}$ that defines the value of a relational expression. To simplify later discussion we shall adopt the following conventions regarding tableaux. The symbols appearing in a tableau are chosen from

(1) Distinguished variables, for which we use a's, possibly with subscripts. These correspond to the symbols to the left of the bar in the set former.

(2) Nondistinguished variables, for which we generally use b's. These are the bound variables appearing in the set former.

(3) Constants, for which we use c's or nonnegative integers.

(4) Blank.

The summary of a tableau may contain only distinguished variables, constants, and blanks. The rows of a tableau may contain variables (distinguished and nondistinguished) and constants. When tableaux represent SPJ-expressions, we can assume that the same variable does not appear in two different columns of a tableau, and that a distinguished variable does not appear in a column unless it also appears in the summary of that column.

Let T be a tableau and let S be the set of all symbols appearing in T (i.e., variables and constants). A *valuation* ρ for T associates with each symbol of S a constant, such that if c is a constant in S, then $\rho(c) = c$. We extend ρ to the summary and rows of T as follows. Let w_0 be the summary of T, and $w_1, w_2, \ldots, w_n$ the rows. Then $\rho(w_i)$ is the tuple obtained by substituting $\rho(v)$ for every variable v that appears in w_i.

A tableau defines a mapping from *universal instances* (relations over the set of all attributes) to relations over a certain subset of the attributes, called the *target relation scheme,* in the following way. If T is a tableau and I an instance, then $T(I)$ is the relation on the attributes whose columns are nonblank in the summary, such that

$$T(I) = \{\rho(w_0) \mid \text{for some valuation } \rho \text{ we have } \rho(w_i) \text{ in } I \text{ for } 1 \leqslant i \leqslant n\}$$

Example 4. Let T be the tableau

A	B	C
a_1	a_2	
a_1	b_1	b_3
b_2	a_2	1
b_2	b_1	b_4

We conventionally show the summary first, with a line below it. We can interpret this tableau as defining the relation on AB

$$T(I) = \{a_1a_2 \mid (\exists b_1)\,(\exists b_2)\,(\exists b_3)\,(\exists b_4) \text{ such that } a_1b_1b_3 \text{ is in } I \text{ and } b_2a_21 \text{ is in } I \text{ and } b_2b_1b_4 \text{ is in } I\}$$

given any universal instance I.

Representation of SPJ-Expressions by Tableaux

We can construct a tableau T to represent any SPJ-expression E in the following manner.

(1) If E is a single relation scheme R, then the tableau T for E has one row and a summary such that:

  (i) If A is an attribute in R, then in the column for A, tableau T has the same distinguished variable in the summary and row.

  (ii) If A is not in R, then its column has a blank in the summary and a new nondistinguished variable in the row.

(2a) Suppose E is of the form $\sigma_{A=c}(E_1)$, and we have constructed T_1, the tableau for E_1.

  (i) If the summary for E_1 has blank in the column for A, then $T = \varnothing$. (We assume $\varnothing$ is a tableau that maps any universal instance into the empty set.)

  (ii) If there is a constant $c' \neq c$ in the summary column for A, then $T = \varnothing$. If $c = c'$, then $T = T_1$.

  (iii) If T_1 has a distinguished variable a in the summary column for A, the tableau T for E is constructed by replacing a by c wherever it appears in T_1.

(2b) Suppose E is of the form $\pi_X(E_1)$, and T_1 is the tableau for E_1. The tableau T for E is constructed by replacing nonblank symbols by blanks in the summary of T_1 for those columns whose attributes are not in X. Distinguished variables in those columns become nondistinguished.

(2c) Suppose E is of the form $E_1 \bowtie E_2$ and T_1 and T_2 are the tableaux for E_1 and E_2, respectively. Let S_1 and S_2 be the symbols of T_1 and T_2, respectively. Without loss of generality, we may take S_1 and S_2 to have disjoint sets of nondistinguished variables, but identical distinguished variables in corresponding columns.

(i) If T_1 and T_2 have some column in which their summaries have distinct constants, then $T = \emptyset$.

(ii) If no corresponding positions in the summaries have distinct constants, the set of rows of the tableau T for E consists of the union of all the rows of T_1 and T_2. The summary of T has in a given column

(a) The constant c if one or both of T_1 and T_2 have c in that column's summary. In this case we also replace any distinguished variable in that column by c.

(b) The distinguished variable a if (a) does not apply, but one or both of T_1 and T_2 have a in that column's summary.

(c) Blank, otherwise.

Example 5. Let A, B and C be the attributes, in that order, and suppose we have the expression $\pi_{AC}(\sigma_{B=0}(AB \bowtie BC))$. By Rule (1), the tableaux for AB and BC are

A	B	C
a_1	a_2	
a_1	a_2	b_1

and

A	B	C
	a_2	a_3
b_2	a_2	a_3

By Rule (2c), the tableau for $AB \bowtie BC$ is

A	B	C
a_1	a_2	a_3
a_1	a_2	b_1
b_2	a_2	a_3

By Rule (2a), the tableau for $\sigma_{B=0}(AB \bowtie BC)$ is

A	B	C
a_1	0	a_3
a_1	0	b_1
b_2	0	a_3

Finally, by Rule (2b), the tableau for $\pi_{AC}(\sigma_{B=0}(AB \bowtie BC))$ is

A	B	C
a_1		a_3
a_1	0	b_1
b_2	0	a_3

Simple Tableaux

A tableau is *simple* if in any column with a repeated nondistinguished variable there is no other symbol that appears in more than one row. For simple tableaux there exists a polynomial time equivalence algorithm, whereas for general tableaux the equivalence problem is NP-complete [3]. It is not easy to find an SPJ-expression with a nonsimple tableau. The expression $\pi_{AC}(AB \bowtie BC) \bowtie (AB \bowtie BD)$ is a minimal expression that gives rise to a nonsimple tableau. The tableau is shown in Fig. 5. The rows in the column for B have repeated nondistinguished and distinguished variables.

A	B	C	D
a_1	a_2	a_3	a_4
a_1	b_1	b_2	b_3
b_4	b_1	a_3	b_5
a_1	a_2	b_6	b_7
b_8	a_2	b_9	a_4

Fig. 5. A nonsimple tableau.

5. Synthesis of Relational Expressions from Tableaux

Let us initially consider a simple tableau T with no constants in the summary, although constants may appear in the rows. Let A be a column in which rows i and j have the same nondistinguished variable, and k a row with a distinguished variable in column A.[†] If T comes from an expression, consider the parse tree P of that expression. Each leaf of P corresponds to an operand. Using the tableau construction algorithm of Section 4, we see that it is possible to make a one-to-one identification of parse tree leaves with rows of the tableau. We shall therefore talk about rows of a tableau as though they were the leaves of the parse tree of a corresponding SPJ-expression.

Suppose that in P, (i, j) is not a proper descendant of (i, k). Since (i, j) and (i, k) cannot be independent, it follows that (i, k) must be a descendant of (i, j). Then consider the tableau constructed for the subexpression rooted at the node (i, k) of P. In particular, which rows have the distinguished variable in column A? Surely k does, as the distinguished variable appears there in tableau T, and in the tableau construction algorithm a nondistinguished variable is never changed into a distinguished variable. If row i does not also have the distinguished variable in column A, then the variables in rows i and j cannot be equated when we create the tableau for the expression represented by node (i, j). However, if both rows i and k have the distinguished variable, then they must have the same variable in T, a contradiction. We can therefore infer that $(i, j) < (i, k)$. Arguing similarly, we can show that $(i, j) < (j, k)$. From these two observations we are led to the conclusion:

[†] In a simple tableau, k will be unique, but we can relax the "simple" constraint to the point that a column with a repeated nondistinguished variable can have a repeated distinguished variable or constant, but not another repeated nondistinguished variable.

(*) Suppose P is the parse tree of an expression yielding tableau T. If rows i and j of T have the same nondistinguished variable in column A, and row k has a distinguished variable in column A, then we must have $(i, j) < (i, k)$ and hence $(i, j) < (j, k)$ in P.

That this is sufficient for simple tableaux is expressed in the next theorem.

Theorem 5. Let T be a tableau having no constants in the summary and no column with two or more repeated nondistinguished variables. Then T comes from an expression if and only if the set of constraints defined in (*) determines a tree.

Proof. *Only if.* We argued above that if T comes from an expression, the constraints (*) must be satisfied.

If. Suppose (*) determines a tree P. We show how to build the parse tree of an expression for T. This expression may use join as a k-ary operator. But since binary join is associative and commutative, nothing is lost by doing so, and the k-ary joins can be replaced by binary joins in any order.

The construction proceeds by induction on the height of a node in the tree P. We shall construct for each subtree an expression yielding a tableau whose rows are the rows of T that are leaves of the subtree but with repeated nondistinguished variables, all of whose instances do not appear among these rows, replaced by distinguished variables. In the theorem the hypothesis about T guarantees that no conflicts arise, that is, no column needs more than one distinguished variable.

Basis. Height 0. The node is a leaf labeled by a relation scheme R, and identified with some row i. Row i cannot have a constant or distinguished variable other than those in columns for attributes in R. (We assume the relation scheme R containing all attributes in which row i has a distinguished variable, constant, or repeated nondistinguished variable can be found. If not all relation schemes are permissible, then it is easy to check for tableaux with illegal rows.)

If row i has constant c in column A, then apply $\sigma_{A=c}$ to R. Then project onto those attributes in whose column row i has either a distinguished variable or a repeated nondistinguished variable. The result is an expression whose tableau has one row, which is row i with repeated nondistinguished variables replaced by distinguished variables.

Induction. We construct the expression for node n from the expressions $E_1, E_2, \ldots, E_m$ representing the children of node n. Begin by joining $E_1, E_2, \ldots, E_m$. Then project onto those attributes A in whose columns either

(1) some descendant leaf of n is a row with a distinguished variable in column A, or

(2) some, but not all, of the instances of the repeated nondistinguished variable in column A are found in the rows whose leaves are descendants of n.

The result is easily seen to satisfy the inductive hypothesis, as in each column A, variables that appear in two or more of the E_i's and that must become the repeated nondistinguished variable for column A are distinguished in the tableaux for the E_i's and are therefore equated in the join.

Finally, the inductive hypothesis applied to the root implies that the constructed expression has tableau T. □

Example 6. Consider the tableau

	A	B	C	D
		a_1	a_2	a_3
1	0	b_1	b_2	b_3
2	b_4	b_1	a_2	b_5
3	b_6	b_7	a_2	a_3
4	b_8	a_1	a_2	b_9

We assume that rows 1, 2, 3 and 4 come from relation schemes (AB), (BC), (CD) and (BC), respectively. The only constraints are $(1,2) < (1,4)$ and $(1,2) < (2,4)$, so we obtain the skeletal tree shown in Fig. 6.

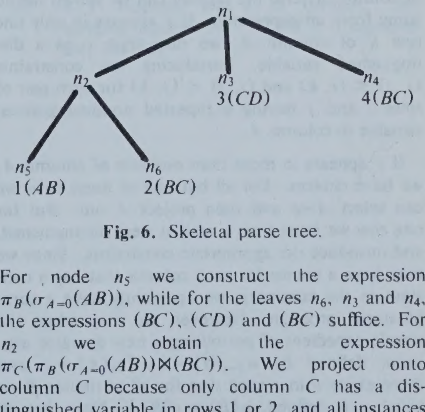


Fig. 6. Skeletal parse tree.

For node n_5 we construct the expression $\pi_B(\sigma_{A=0}(AB))$, while for the leaves n_6, n_3 and n_4, the expressions (BC), (CD) and (BC) suffice. For n_2 we obtain the expression $\pi_C(\pi_B(\sigma_{A=0}(AB))\bowtie(BC))$. We project onto column C because only column C has a distinguished variable in rows 1 or 2, and all instances of the repeated nondistinguished variable b_1 are found in these rows. The expression for n_1 is obtained by joining the above expression with (CD) and (BC) in any order. We should then project onto B, C, and D, but this projection is seen to be superfluous, since the join produces a relation over only those attributes. The final expression is shown in Fig. 7. □

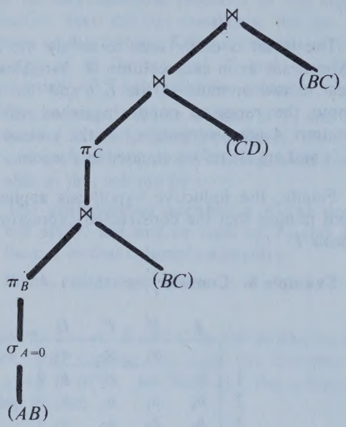


Fig. 7. Synthesized expression.

6. Extension to the Case Where Constants Appear in the Summary

If there is a constant c in column A of the summary of a tableau, then c must appear in some row of column A, else the tableau can be shown not to come from an expression. If c appears in only one row k of column A, we may treat c as a distinguished variable, introducing the constraints $(i, j) < (i, k)$ and $(i, j) < (j, k)$ for each pair of rows i and j having a repeated nondistinguished variable in column A.

If c appears in more than one row of column A, we have choices. For all but one of these rows we can select $A=c$ and then project A out. But for one row we must treat c as if it were distinguished, and introduce the appropriate constraints. Since we may have a choice for each column that has a constant in the summary, we apparently have a combinatorial problem. However, we may adopt the simple expedient of permitting a new operator *augment*, defined by $\alpha_{A=c}(r) = \{ \mu \mid \mu(A) = c$ and there exists ν in r such that for all attributes B on which r is defined, $\mu(B) = \nu(B) \}$. Note that we may assume that r is not defined on A, otherwise augment is the same as select. Then we may synthesize an expression from a tableau with constants in the summary by:

(1) deleting constants from the summary,

(2) synthesizing an expression for the resulting tableau, if it exists, and then

(3) using the augment operator to introduce the constants into the tuples of the relation resulting from application of the expression from (2).

7. Finding an Expression Equivalent to a Given Tableau

We should also consider a variant of the problem of finding an expression that yields a given tableau. In most circumstances it will be sufficient to find an expression that yields an equivalent tableau, that is, an expression defining the same mapping from universal instances to target relations as the tableau.

It turns out that this question is no harder than the original problem, as we can show that a tableau comes from an expression if and only if its minimal equivalent tableau does. Thus as a bonus, we can even obtain an expression with the fewest joins equivalent to a given tableau, if any equivalent expression exists.

Theorem 6. If a tableau comes from an expression, then its minimal equivalent tableau comes from an expression.

Proof. It follows from [4] that any given tableau T has a minimal equivalent tableau T' (one with the fewest number of rows) such that each row of T' is a row of T. Given an expression E yielding tableau T, we can delete the operands of E corresponding to rows of T that are not in T'.

Nodes that apply unary operators (select and project) to a deleted operand, and nodes that join two deleted operands are themselves deleted. Nodes that join a deleted operand with one that is not deleted are identified with the non-deleted operand. The resulting expression will yield tableau T'. We may prove by an easy induction on the height of a node remaining in the expression that the tableau for this node is the same as the tableau for the corresponding node in E, with rows that do not eventually become rows in T' deleted. □

Example 7. The tableau in Fig. 8

	a_1	a_2	a_3	
1	a_1	b_1	b_2	b_3
2	b_4	b_1	b_5	b_6
3	a_1	a_2	b_7	b_6
4	b_8	a_2	a_3	b_9

Fig. 8. A tableau.

implies constraints $(1,2) < (1,3)$, $(1,2) < (2,3)$, $(1,2) < (1,4)$, and $(1,2) < (2,4)$, so we may synthesize the expression of Fig. 9.

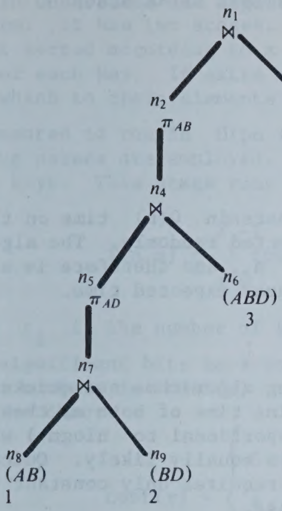


Fig. 9. An expression for the tableau of Fig. 8.

The minimal equivalent tableau for Fig. 8 has only rows 3 and 4. To synthesize an expression for the minimal tableau, we delete nodes n_8 and n_9, which correspond to rows 1 and 2. This causes nodes n_7 and n_5 to be deleted and node n_4 to be merged with n_6. The resulting expression is shown in Fig. 10.

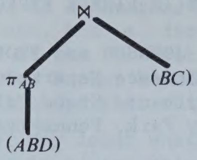


Fig. 10. Expression for minimal tableau.

References

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.

[2] A. V. Aho, Y. Sagiv, and J. D. Ullman, "Efficient optimization of a class of relational expressions," *Proc. ACM SIGMOD International Conference on the Management of Data*, Austin, Texas, 1978.

[3] A. V. Aho, Y. Sagiv, and J. D. Ullman, "Equivalences Among Relational Expressions," *SIAM J. Computing*, to appear.

[4] A. K. Chandra and P. M. Merlin, "Optimal Implementation of Conjunctive Queries in Relational Databases," *Proc. Ninth Annual ACM Symposium on Theory of Computing*, May 1976, pp. 77-90.

[5] E. F. Codd, "A Relational Model for Large Shared Data Banks," *Comm. ACM* 13:6 (June, 1970), 377-387.

[6] J. E. Hopcroft and J. D. Ullman, "Set merging algorithms," *SIAM J. Computing* 2:4, 294-303.

SORTING NUMBERS IN LINEAR EXPECTED TIME AND CONSTANT EXTRA SPACE[†]

DONALD B. JOHNSON and TEOFILO F. GONZALEZ
Computer Science Department
The Pennsylvania State University
University Park, Pennsylvania 16802

ABSTRACT

An algorithm is shown which sorts n numbers in O(n) time on the average where the numbers to be sorted are selected randomly. The algorithm uses only constant extra space, independent of n , and therefore is an advance over known algorithms which sort in linear expected time.

INTRODUCTION

Among the most widely used internal sorting algorithms are quicksort [2,3,4] and heapsort [8,4]. The expected running time of both of these algorithms is $\theta(n\log n)$ (time asymptotically proportional to $n\log n$) when sorting n numbers each permutation of which is equally likely. Quicksort consumes $O(\log n)$ extra space whereas heapsort requires only constant extra space. Both space bounds apply in the worst case.

In 1965, MacLaren [5] presented an algorithm which sorts numbers in $\theta(n)$ time in the expectation when the input numbers are independently and uniformly distributed. We note that this randomness assumption implies each permutation equally likely, but the converse does not hold. One reason, perhaps, that this result has received little attention is that MacLaren's algorithm requires for some fixed integer $p \geq 1$ at least $n + n^{1/p}$ extra locations which in general must be as large as those required for the numbers to be sorted. Variants proposed by others also require asymptotically significant extra space. Knuth's improvement [4] needs at least $\sqrt{n}$ extra locations. Dobosiewicz's recursive version of MacLaren's algorithm [1] requires at least 2n extra locations.

Our algorithm sorts n numbers in $\theta(n)$ expected time under the randomness assumption, and it requires only constant extra space independent of n . In addition to these asymptotic results we also discuss evidence below which indicates our algorithm could be implemented to perform slightly faster than quicksort on files of size practical for internal sorting. It has been asserted (see [6], for instance) that quicksort is the best extant sorting algorithm for practical purposes, so an estimate of comparative running times on files of practical size is of some interest.

In order to determine the extra space required by our algorithm it is necessary to assume that the registers, or memory locations, which hold a single input number, or key, have a fixed size which in general is necessary to represent the key. Under this assumption it is assumed impossible to encode additional information in a location containing a key without loss of information in the key. To make this assumption concrete, let each location containing a key be capable of storing a binary number of at most h bits and let the keys be h-bit nonnegative numbers. Our results could easily be restated under other assumptions for they are in fact independent of the question of number representation provided arithmetic operations on numbers of magnitude adequate to represent keys are taken to be of unit cost.

[†] This work was partially supported under NSF grant MCS77-21092.

To understand our algorithm it is helpful to understand MacLaren's algorithm. It has two stages. In the first stage, the n input keys are bucket sorted according to a subkey comprised of the m most significant bits of each key. If extra space is allowed for bucket headers and pointers with which to chain elements in buckets, it is clear that this stage can be implemented to run in $O(pn + p2^{m/p})$ time for any input where $p \geq 1$ bucket sorting passes are employed. The second stage is an insertion sort on the whole keys. This stage runs in time asymptotically proportional to

$$T(n,m) = \sum_{i=0}^{2^m-1} \mathrm{cost}(r_i)$$

where r_i is the number of keys for which the subkeys comprised of the m most significant bits have value i and the cost to fully sort a sequence of y numbers is $\mathrm{cost}(y)$. To reflect the asymptotic cost of insertion sort, we choose

$$\mathrm{cost}(y) = \begin{cases} 0 & y=0\ , \\ 1 & y=1\ , \\ y(y-1) & y=2,\ldots,n\ . \end{cases}$$

Under the randomness assumption the probability that $r_i=k$ is

$$\binom{n}{k}\left(\frac{1}{2^m}\right)^k\left(1-\frac{1}{2^m}\right)^{n-k}$$

for $k=0,\ldots,n$. Thus, for $i=0,\ldots,2^m-1$,

$$E(T(n,m)) = E\left(\sum_{i=0}^{2^m-1} \mathrm{cost}(r_i)\right) = \sum_{i=0}^{2^m-1} E(\mathrm{cost}(r_i)) = 2^m E(\mathrm{cost}(r_i)),$$

and

$$E(\mathrm{cost}(r_i)) = \sum_{k=0}^{n} \mathrm{cost}(k)\binom{n}{k}\left(\frac{1}{2^m}\right)^k\left(1-\frac{1}{2^m}\right)^{n-k}$$

$$= \frac{n}{2^m}\left(1-\frac{1}{2^m}\right)^{n-1} + \sum_{k=2}^{n} k(k-1)\binom{n}{k}\left(\frac{1}{2^m}\right)^k\left(1-\frac{1}{2^m}\right)^{n-k}$$

$$= \frac{n}{2^m}\left(1-\frac{1}{2^m}\right)^{n-1} + \frac{n(n-1)}{2^{2m}}\sum_{k=2}^{n}\binom{n-2}{k-2}\left(\frac{1}{2^m}\right)^{k-2}\left(1-\frac{1}{2^m}\right)^{n-k}$$

$$= \frac{n}{2^m}\left(1-\frac{1}{2^m}\right)^{n-1} + \frac{n(n-1)}{2^{2m}}\ .$$

Substituting,

$$E(T(n,m)) = n\left(1 - \frac{1}{2^m}\right)^{n-1} + \frac{n(n-1)}{2^m} .$$

To obtain linear running time for this stage MacLaren sets $m=\log_2 n$ for n a power of two. If we choose $m=\lceil \log_2 n \rceil$ then

$$E(T(n,\lceil \log_2 n \rceil)) \leq \frac{4n}{3\sqrt{e}} + n - 1 = O(n), \qquad n>1 .$$

MacLaren proposed $p=2$ in stage 1 but it is clear that any constant number of passes $p \geq 1$ will yield an algorithm with $O(n)$ running time overall provided $m=\overline{\theta}(\log n)$. Knuth's modification alluded to above is MacLaren's algorithm with $p=2$ in which the bucket sort is replaced by an address computation sort, thereby eliminating the pointers used to chain keys within buckets. Extra space is still consumed by the $\sqrt{n}$ locations needed for bucket headers. The algorithm reported by Dobosiewicz is MacLaren's algorithm with $p=1$, but it applies stage 1 of MacLaren's algorithm recursively in place of the stage 2 insertion sort. In addition, a linear time median finding algorithm is used to balance the number of elements between the first $\lfloor n/2 \rfloor$ buckets and the remaining buckets. This latter feature improves the worst case running time to $O(n\log n)$ while worsening the expected running time by a constant factor.

SORTING IN LINEAR EXPECTED TIME AND CONSTANT EXTRA SPACE

Insertion sort requires only constant extra space. Thus to present a linear expected time sorting algorithm which runs in constant extra space it suffices to show how to implement a one-pass bucket sort on large keys, the magnitude of which is of the order of the number of keys, which runs in linear time and no more than constant extra space.

Given an array X of keys, the array being indexed from L to R, the algorithm shown below permutes these keys so that they are sorted in nondecreasing order on the subkeys $k(i)$, which are comprised of $m=\lceil \log_2(R-L+1) \rceil$ bits of $X(i)$ for $i=L,\ldots,R$. We assume the following definitions for the variables describing the several fields of key $X(i)$ which the algorithm references. If the key $X(i)$ has h bits numbered from 0 through $h-1$, then for $i=L,\ldots,R$,

$$\begin{aligned} code(i) &= X(i)[0,1] \\ k(i) &= X(i)[2,m+1] \\ z(i) &= X(i)[m+2,h] \\ y(i) &= X(i)[2,h] \end{aligned}$$

where $X(i)[a,b]$ for $a \leq b$ denotes the field composed of bits a through b of $X(i)$. See Figure 1. It is assumed that h satisfies $h \geq m+2$. Alternatively we can express $X(i)[a,b]$ as

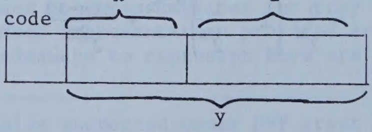


Figure 1. Fields of key X .

$$\left\lfloor \frac{\underline{\mathrm{rem}}(X(i),2^{h-a})}{2^{h-b-1}} \right\rfloor$$

We assume code(i)=0 for i=L,...,R . We will also use a bitwise notation, ie. code(i)='00' . In addition we assume X(i)[2,2] , the leftmost bit in k(i), satisfies X(i)[2,2]=0 , for i=L,...,R , so it is possible to count up to R-L in any k(i) while restricting $|I| \le R-L+1$ for $I = \{k(i) | i=L,\ldots,R\}$, the set of distinct subkeys k . We obtain from arbitrary input X an input satisfying the restrictions on the first three bits of each key by sorting X on X(i)[0,2] for i=L,...,R and recording the positions of the eight subsequences in the resulting X which are uniform on these fields. Thus these fields may be zeroed, to be restored at the end of the computation.

The algorithm which runs under these assumptions is as follows. The phases in which more than one loop traverses X can be condensed to a single loop in each case. We have decomposed these phases to aid in understanding the correctness proof.

```
//Phase 1:  For each distinct key, move one instance to corres-
            ponding address, mark and count.  //

for  i←L to R  do
   t ← k(i)+L
   while  k(t)≠t-L  do
      X(i) ↔ X(t)
      t ← k(i)+L
   endwhile
endfor

for  i←L to R  do
   if  k(i)+L=i  then [ code(i) ← '01'
                        k(i) ← 0 ]
endfor

for  i←L to R  do
   if  code(i)='00'  then [ k(k(i)+L) ← k(k(i)+L)+1 ]
endfor

//Phase 2:  Set initial positions for the distinct keys and mark
            their initial positions.  //

t ← R+1-L
for  i←R to L by -1  do
   if  code(i)='01'  then [ t ← t-k(i)-1
                            k(i) ← t ]
endfor

for  i←L to R  do
   if  code(i)='01' or code(i)='11'  then  [ code(k(i)+L) ←
                                             code(k(i)+L)or'10' ]
endfor

//Phase 3:  Set final positions for all keys.  //

for  i←L to R  do
   if  code(i)='10' or code(i)='00'  then  [ k(k(i)+L) ←
                                                  k(k(i)+L)+1
                                             k(i) ← k(k(i)+L)-1 ]
endfor
```

```
   //Phase 4:  Rearrange.  //
27 for  i←L to R  do
28    while  k(i)≠i-L  do
29       y(i) ↔ y(k(i)+L)
30    endwhile
31 endfor
   //Phase 5:  Substitute subkey values and replace codes.  //
32 i ← j ← L
33 loop
34    while  code(i)='00' or code(i)='10'  do
35       i ← i+1
36    endwhile
37    repeat
38       k(j) ← i-L
39       j ← j+1
40       if  j>R  then  [ exit loop ]
41    until  code(j)='10' or code(j)='11'
42    i ← i+1
43 forever
44 for  i←L to R  do
45    code(i) ← '00'
46 endfor
```

Before proving correctness of our algorithm shown above we state the following definitions. Let $\hat{X}$ be any sorted order of X . Then, for $j \epsilon I$, ,

$$s(j) = \min_{L \le i \le R} \{i-L \mid \hat{k}(i)=j\} ,$$

$$f(j) = \max_{L \le i \le R} \{i-L \mid \hat{k}(i)=j\} , \text{ and}$$

$$c(j) = f(j) - s(j) + 1 .$$

Furthermore, let X^0 be a copy of X just before phase 1 is executed.

LEMMA. Let X^0 and X be as defined above. There exist π_1 and π_2 , permutations of $\{0,\ldots,R-L\}$, for which the following conditions hold after execution of phases 1 through 5:

(i) $X(\pi_2(\pi_1(j))+L) = X^0(j+L)$, $j=0,\ldots,R-L$, and

(ii) $k(j+L) \le k(j+L+1)$, $j=0,\ldots,R-L-1$.

Furthermore, the algorithm terminates following phase 5.

Proof: It can be verified that all phases terminate. Therefore let X^1 be a copy of the program variable X following termination of the loop ending at line 7. Since the only operations on X are exchanges of elements,

$$X^1(\pi_1(j)+L) = X^0(j+L) , \qquad j=0,\ldots,R-L ,$$

where π_1 is a permutation of $\{0,\ldots,R-L\}$. It can be seen that X^1 satisfies $k^1(j+L)=j$ for $j \epsilon I$. Thus, to complete the proof that the algorithm correctly sorts the input it remains to be shown that there exists, upon termination of phase 5, a permutation π_2 of $\{0,\ldots,R-L\}$ for which

$$\text{(i)}\quad X(\pi_2(j)+L) = X^1(j+L)\ , \qquad j=0,\ldots,R-L\ , \text{ and}$$

$$\text{(ii)}\quad k(j+L) \leq k(j+L+1)\ , \qquad j=0,\ldots,R-L-1\ .$$

The remainder of phase 1 sets code(j+L)='01' and k(j+L)=0 for $j \varepsilon I$ in lines 8 through 11 and then counts distinct keys in k(j+L) for $j \varepsilon I$ yielding these conditions when phase 1 terminates:

$$z(j+L) = z^1(j+L)\ , \qquad j=0,\ldots,R-L\ ,$$

$$k(j+L) = \begin{cases} c(j)-1\ , & j \varepsilon I\ , \\ k^1(j+L)\ , & \text{otherwise, and} \end{cases}$$

$$\text{code}(j+L) = \begin{cases} \text{'01'}\ , & j \varepsilon I\ , \\ \text{'00'}\ , & \text{otherwise}\ . \end{cases}$$

It can be verified that the information retained in X is sufficient to restore X^1 and therefore to complete the sort.

In phase 2, the loop comprised of lines 15 through 19 sets k(j+L) to s(j) for $j \varepsilon I$. The loop at lines 20 through 22 sets code(s(j)+L) to code(s(j)+L) or '10' for $j \varepsilon I$. Consequently X^2, the result of phase 2, has the properties

$$z^2(j+L) = z^1(j+L)\ , \qquad j=0,\ldots,R-L\ ,$$

$$k^2(j+L) = \begin{cases} s(j)\ , & j \varepsilon I\ , \\ k^1(j+L)\ , & \text{otherwise, and} \end{cases}$$

$$\text{code}^2(j+L) = \begin{cases} \text{'q1'}\ , & \text{iff } j \varepsilon I\ , \\ \text{'1q'}\ , & \text{iff } j=s(\ell) \text{ for some } \ell \varepsilon I\ , \end{cases}$$

where $q \varepsilon \{0,1\}$. Again it can be verified that X contains information sufficient to restore X^1 .

The assignments in phase 3 are executed for those values of i for which $i-L \not\varepsilon I$. The effect is to set k(j+L) to $\pi_2(j)$, for j=0,...,R-L, for some π_2 which will sort X^1 . Thus each field k(j+L) will encode the position of $X^1(j+L)$ in the final sort. We support this assertion by considering one fixed value $t \varepsilon I$. Since there are f(t)-s(t) distinct values of i for which $i-L \not\varepsilon I$ and $k^1(i)=t$, there will occur a sequence of assignments induced by these values, which we denote $\{i_1,\ldots,i_{f(t)-s(t)}\}$, as follows.

$$\begin{aligned} k(t+L) &\leftarrow k(t+L)+1 = s(t)+1 \\ k(i_1) &\leftarrow k(t+L)-1 = s(t) \\ k(t+L) &\leftarrow k(t+L)+1 = s(t)+2 \\ k(i_2) &\leftarrow k(t+L)-1 = s(t)+1 \\ &\vdots \\ k(t+L) &\leftarrow k(t+L)+1 = f(t) \\ k(i_{f(t)-s(t)}) &\leftarrow k(t+L)-1 = f(t)-1 \end{aligned}$$

where $\{X^1(i_1),\ldots,X^1(i_{f(t)-s(t)}),X^1(t+L)\}$ is precisely the set of keys which must be in positions $s(t)+L$ through $f(t)+L$ in any sorted order $\hat{X}$ of X^1. Furthermore this sequence of assignments references locations disjoint from the locations referenced for any $t'\neq t$, $t'\varepsilon I$. Thus it can be seen that the sequence $k^3(L),\ldots,k^3(R)$ records a suitable permutation π_2 where X^3 is a copy of X after phase 3 is completed.

With this understood, the operation of phase 4 is clear. It simply applies π_2 to y^3 (X^3 exclusive of the code fields). The code fields still record the entire structure of $\hat{k}(j)$ for $j=L,\ldots,R$ for any sorted order $\hat{X}$ of X^1 (or X^0 for that matter). For X after phase 4, then, these facts may be stated precisely as

$$z(\pi_2(j)+L) = z^1(j+L)\ , \qquad j=0,\ldots,R-L\ ,$$

$$\text{code}(j+L) = \begin{cases} \text{'q1'}\ , & \text{iff } j\varepsilon I\ , \\ \text{'1q'}\ , & \text{iff } j=s(\ell) \text{ for some } \ell\varepsilon I\ , \end{cases}$$

where $q\varepsilon\{0,1\}$. The values in k are now immaterial since the code fields for the entire array encode what each k value should be.

Phase 5 completes the k fields by extracting this information. Index i is advanced until $i=\min_{\ell\varepsilon I}\{\ell+L\}$, indicating that $i-L$ is the first distinct subkey k in X^0. Index j is then incremented from $s(i-L)$ to $f(i-L)$ in order to store $i-L$ as needed. Repetition for each $i-L\varepsilon I$ yields, for X after line 43,

$$z(\pi_2(j)+L) = z^1(j+L)\ , \qquad j=0,\ldots,R-L\ ,$$

$$k(\pi_2(j)+L) = k^1(j+L)\ , \qquad j=0,\ldots,R-L\ , \text{ and}$$

$$k(j+L) \leq k(j+L+1)\ , \qquad j=0,\ldots,R-L-1\ .$$

When phase 5 resets all code fields to '00' we then have

$$X(\pi_2(j)+L) = X^1(j+L)\ , \text{ or}$$

$$X(\pi_2(\pi_1(j)+L)) = X^0(j+L)\ , \qquad j=0,\ldots,R-L\ , \text{ and}$$

$$k(j+L) \leq k(j+L+1)\ , \qquad j=0,\ldots,R-L-1\ ,$$

as required. □

THEOREM. It is possible to sort n numbers in $O(n)$ expected time and constant extra space provided the numbers are identically and uniformly distributed and at least $h=\lceil\log_2 n\rceil+c$ bits are required to represent the numbers, for any fixed constant c.

Proof: For $c\geq 2$ a complete algorithm is:

(1) Bucket sort X on three most significant bits.

(2) Sort each subsequence of X produced in stage (1) using the algorithm described above.

(3) Insertion sort X.

It is well known how to implement stage (1) to run in $O(n)$ time and to use only constant extra space. It has been established in the lemma given above that stage (2) can be performed using only constant extra space. The procedure exhibited for stage (2) can easily be seen to run in $O(n)$ time when it is noticed that there can be no more than n iterations in total for each of the while loops at lines 3 and 28. Given the analysis in the introduction, it is clear that stage (3) runs in $O(n)$ expected time. Of course, it is well known how to implement insertion sort to use only constant extra space.

Whenever $c<2$ at most 2^{2-c} auxiliary counters are required. ☐

The above theorem can be shown to hold for any h whenever n is divisible by the number of distinct keys. The construction involves grouping equal keys in adjacent locations to expand counting capacity. We omit the details. Of course, the theorem holds when h is a constant in any event.

CONCLUSION

We do not consider whether this algorithm is practical in the sense that it should be used in data processing. However, the recent analysis of quicksort by Sedgewick [6,7] invites an attempt to estimate at what value of n does $L(n)=Q(n)$ where $L(n)$ is the running time of our linear time algorithm and $Q(n)$ is the running time of quicksort. Certainly if this n were very large, our tacit assumption that our algorithm can reasonably be called an internal sort would be brought into question.

Sedgewick employs a computational model which essentially counts memory references on an idealized computer in some implementation at the memory reference level. Although our algorithm is susceptible to a similar analysis, we have taken a different approach. Using the second paper cited above we have produced an implementation of quicksort on the IBM 370 which appears to admit only minor improvements. Then the task has been to implement our algorithm as well as we can and call the n at which the running times are equal, an estimate for an upper bound on the crossover. For simplicity we choose the number of machine instructions executed as our time measure.

We have an implementation of a more complicated algorithm, a predecessor of the one presented in the previous section, for which our crossover estimate is $n=200{,}000$. Preliminary results on this new version indicate that a value of n between 20,000 and 50,000 can easily be demonstrated above which $L(n)<Q(n)$.

It is not likely that our algorithm can be speeded up by replacing insertion sort, in the last stage, with another sorting method. This judgment follows from observing that $E(r_i>0)$, the expected length of subsequences which the final sort must reorder, is small. This quantity is given by

$$E(r_i>0) = \sum_{k=1}^{n} k\frac{\text{Prob}(r_i=k)}{\text{Prob}(r_i>0)}$$

$$= \sum_{k=1}^{n} k\binom{n}{k}\left(\frac{1}{2^m}\right)^k \left(1-\frac{1}{2^m}\right)^{n-k} \Big/ \left(1-\binom{n}{0}\left(1-\frac{1}{2^m}\right)^n\right)$$

$$= \frac{n}{2^m\left(1 - \left(1 - \frac{1}{2^m}\right)^n\right)}$$

which for $m=\lceil \log_2 n \rceil$ evaluates asymptotically to less than or equal to

$$\frac{1}{1 - \frac{1}{e}} = \frac{e}{e - 1} \simeq 1.58 \ .$$

It is easy to see that our algorithm is not stable (see [4]). However, neither are quicksort, heapsort, Knuth's modification of MacLaren's algorithm, nor Dobosiewicz's algorithm. It is an open question whether there exists a stable sorting algorithm which runs in O(n) expected time and constant extra space.

REFERENCES

1. W. Dobosiewicz, Sorting by distributive partitioning, Inf. Proc. Letters 7, 1(Jan. 1978) 1-6.
2. C. A. R. Hoare, Partition(Algorithm 63); Quicksort(Algorithm 64); and Find(Algorithm 65), Comm. ACM 4, 7(July 1961) 321-322.
3. C. A. R. Hoare, Quicksort, Computer J. 5, 4(April 1962) 10-15.
4. D. E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, 1973.
5. M. D. MacLaren, Internal sorting by radix plus sifting, J. ACM 13, 3 (July 1966) 404-411.
6. R. Sedgewick, The analysis of quicksort programs, Acta Informatica 7 (1977) 327-355.
7. R. Sedgewick, Implementing quicksort programs, Tech. Report No. CS-38, Brown University (March 1977), (to appear Comm. ACM).
8. J. W. J. Williams, Algorithm 232: Heapsort, Comm. ACM 7, 6(1964) 347-348.

HAMILTONIAN PATHS ON A RECTANGULAR CHESSBOARD

F. LUCCIO* and C. MUGNAI**
*University of Pisa, Italy, **University of Florence,Italy

ABSTRACT

As part of the problem of traversing a two dimensional data structure, we discuss how to draw a Hamiltonian path of given extremes on a rectangular chessboard R, where single steps parallel to the sides of R are the only allowed movements.

Rectangles with one side of length ≤ 3 are treated as special cases, where the existence of a Hamiltonian path p is related to the value of the distance of the extremes of p from the vertices of R. In the general case, a Hamiltonian path p can always be drawn for rectangles with an even number of cells, while strict conditions on the positions of the extremes of p must be met for rectangles with an odd number of cells.

INTRODUCTION

We work on a two dimensional structure consisting of an infinite chessboard, on which we define rectangular figures, and study how to traverse such figures. Each elementary cell C is adjacent to the four surrounding cells sharing one edge with C, and the only allowed movements (steps) are between adjacent cells. (That is, we work in L_1 metric). This problem arises in the organization of data in two dimensional memory supports, such as array type magnetic bubble memories, where a read/write device can be moved in two perpendicular directions in the memory plane [1]. In this case, the decomposition of any set of data into rectangles is a spontaneous starting point in information retrieval applications, where data tend to cluster in compact patterns [2]. Traversing rectangles is therefore important. If data are instead scattered in the memory, the problem can be approached as proposed in [3].

A step from cell A to cell B is denoted by A:B. A path p from cell A to cell B, denoted by $p = A::B$, is a sequence of steps connecting A and B, such that no cell in p is encountered more than once. A Hamiltonian path (h-path) is one traversing all the cells of a given figure. Similarly, a cycle is a closed path, and a Hamiltonian cycle (h-cycle) is a closed h-path.

Let d(A,B) indicate the rectilinear distance (in L_1 metric) between cells A and B. A preliminary fact, valid for any connected figure F of N cells, on the infinite chessboard, is the following.

Fact 1. Any h-path $h = A::B$ on F satisfies:
d(A,B) even, if N is odd;
d(A,B) odd, if N is even.

The proof follows from the observation that the number of steps of h has the same parity degree of d(A,B), and it must be equal to N-1.

Two cells A,B satisfying the conditions of fact 1 are said to be compatible (with F).

The aim of this work is to build h-paths and h-cycles for any given rectangle. This specific problem seems not to have been treated before, although so many studies on Hamiltonian paths exist (see for example [4]).

H-PATHS OF GIVEN EXTREMES

Let R be a rectangle; x and y be the lengths (i.e. number of cells) of its sides; $N = x \cdot y$. We occasionally write R(x,y) for R, and N(R) for N. To avoid trivial cases, we always assume that $x > 1$, $y > 1$. We now determine conditions under which a h-cycle, or a h-path of given compatible extremes, exists on R, and show how to build such a path. We distinguish among several cases, the simplest of which are treated in this section. Our results are presented as "facts". For brevity all formal proofs have been omitted, and only the basic lines of such proofs are indicated. The reader may complete the proofs by himself, or may refer to [5].

A very simple result holds for h-cycles.

Fact 2. A h-cycle exists on R if and only if N is even.

The if part is proved by the existence of the h-cycle in figure 1. The only if part follows from the observation that the number of steps of a h-cycle must be even, and it must be equal to N.

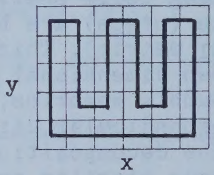
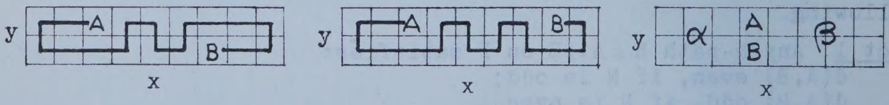


Fig. 1. h-cycle for N even ($x = 6$).

Let us now turn to h-paths, starting from rectangles with a side of length two.

Fact 3. Let $y = 2$, $x \geqslant 2$. For any pair of compatible cells A,B a h-path $h = A::B$ always exists, unless A and B are as in figure 2.c.

In fact, there are essentially three distinct cases for the positions of A,B, indicated in figure 2. A h-path $h = A::B$ can be built only in the cases of figure 2.a and 2.b.



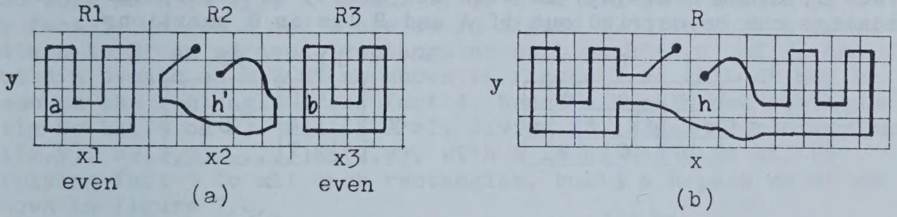
(a) (b) (c)

Fig. 2. (a),(b): $h = A::B$ for $y = 2$. (c): The portions α and β are non void, hence no h-path $h = A::B$ exists.

To study the h-path problem on rectangles with $x,y > 2$, we now present a preliminary result. Let a rectangle $R(x,y)$, $x,y>2$, be decomposed in three adjoining rectangles $R1(x1,y)$, $R2(x2,y)$, $R3(x3,y)$, with $x = x1 + x2 + x3$ and $x1$, $x3$ even (figure 3a). We have:

Fact 4. If a h-path h' exists on R2, then a h-path h exists on R.

To prove this fact, use fact 2 to build two h-cycles a,b on R1,R3 respectively, and observe that, since $y > 2$, there is at least one couple of adjacent steps in a and h', and one couple in h' and b. Therefore, build a h-path h as shown in figure 3.b.



(a) (b)

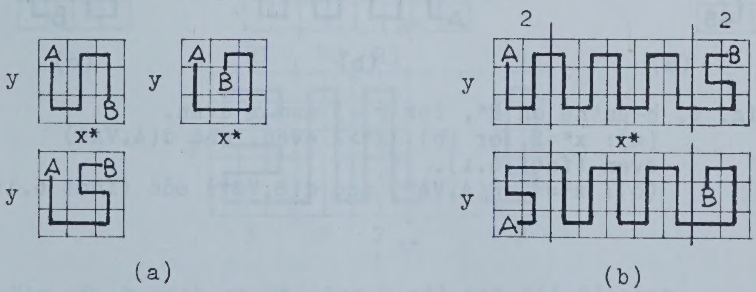
Fig. 3. Illustration of fact 4.

From fact 4 we derive a reduction technique for building a h-path $h = A::B$ on R. Namely, if R can be decomposed into R1, R2, R3 as in fact 4, with R1 or R3 possibly void, and A,B lie on R2, the problem is reconducted to finding a h-path $h'=A::B$ on R2. If more than one such a decomposition exists, consider the one with minimum value of x2, such that $x2 \geqslant 2$. In this case, R2 is called the core of R, denoted by $R^*=R^*(x^*,y)$, where $x^*=x2$. We will generally work on R^*.

To study rectangles with a side of length three, we need to consider the distances of a cell of R from the four cells lying at the vertices of R (briefly, vertices). If N is odd, all such four distances are even, or all of them are odd. If N is even, two such distances are even, and the other two are odd.

Fact 5. Let $y = 3$, $x \geqslant 3$ and x odd. For any compatible pair A,B there exists a h-path $h = A::B$, if and only if $d(A,V)$ is even, where V is any vertex of R.

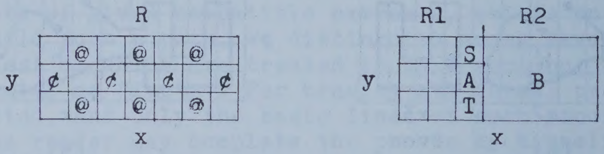
(Note that N is odd, then $d(A,B)$ is even by fact 1. Hence, if $d(A,V)$ is even, $d(B,V)$ is also even). To prove the if part, operate on R^* as shown in figure 4. Remember that $x = x1+x^*+x3$, with x1 and x3 even. Hence x^* must be odd and $d(A,V^*)$ must be even, where V^* is any vertex of R^*.



(a) (b)

Fig.4. h-paths on R^*. $y=3$, $d(A,V^*)$ even. (a): $x^*=3$. (b): $x^*>3$ odd.

To prove the only if part, we let d(A,V) be odd, and prove that no h-path h exists. Since A and B can occupy the positions indicated by @ and ¢ in figure 5.a, several cases should be examined. Let us study one of them, namely, A (or B) is in a ¢ position. Divide R in two rectangles R1,R2 as shown in figure 5.b, with A in R1 and B in R2. i) If h leaves A to enter R1-{A}, then it must cover R1-{A} before entering R2 from cell S or T. This contradicts fact 1, since N(R1) is odd and d(A,S) and d(A,T) are odd. ii) If h leaves A to enter R2, then it must traverse R1-{A} from S to T, or vice versa. This contradicts fact 1, since N(R1-{A}) is even and d(S,T) is even. Similar reasonings can be carried out if A and B lie in @ positions.



(a) (b)

Fig. 5. No h-path exists if $y = 3$, x odd and d(A,V) odd.
(a): Positions for A and B. (b): A in a ¢ position.

The study of h-paths for $y = 3$ and x even is directly conducted on the core R*(x*,y).

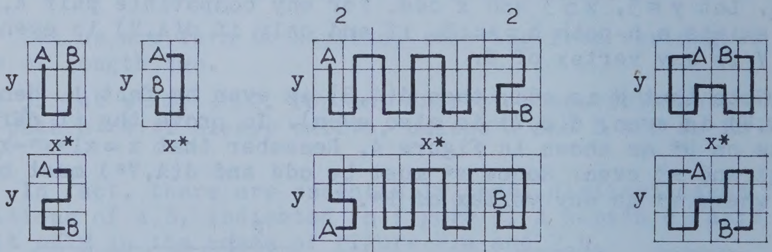
Fact 6. Let $y = 3$, $x \geqslant 4$ and x even. For any compatible pair A,B there exists a h-path $h = A::B$ if and only if:

i. d(A,VA*) or d(B,VB*) is even, where VA* and VB* are the vertices of R* closest to A and B, respectively;

or

ii. x*=4, and A and B lie on the sides of R* of length x*.

To prove the if part we operate on R*, as shown in figure 6. The only if part can be proved similarly to the one of fact 5.



(a) (b) (c)

Fig. 6. h-paths on R*, for $y = 3$ and x even.
(a): x*=2, or (b): x*>2 even, and d(A,VA*) even (fact 6.i).
(c): x*=4, d(A,VA*) and d(B,VB*) odd (fact 6.ii).

GENERAL CONDITIONS FOR THE EXISTENCE OF H-PATHS

We now present our results on the existence of h-paths on rectangles of generic dimensions, and show how to build such paths whenever possible.

Fact 7. Let $x > 3$, $y > 3$, N even. For any compatible pair A, B there exists a h-path $h = A::B$.

To prove fact 7 we assume, without loss of generality, that x is even (note that N is even, then x and/or y must be even), and work on $R^*(x^*, y)$, x^* even. i) If $x^*=2$ a h-path exists on R^* by fact 3, unless A and B are as in figure 2.c. In such a case, attach to R^* an adjacent rectangular portion $R'(2,y)$ of R, and build a h-path on $R' \cup R^*$ as shown in figure 7.a. ($R' \cup R^*$ may be seen as the rectangle R2 of fact 4, hence a h-path can subsequently be built on R). ii) If $x^*>2$, divide R^* in $n=x^*/2$ rectangles $R1(2,y)$, $R2(2,y)$,...., $Rn(2,y)$, with A in R1 and B in Rn. By applying fact 3 to all such rectangles, build a h-path on R^* as shown in figure 7.b.

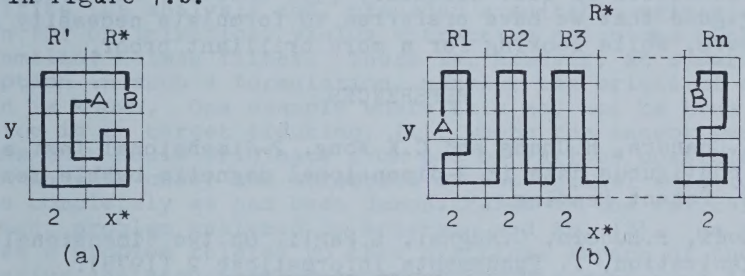


Fig. 7. h-path on $R^* \cup R'$ (a), or on R^* (b), for N even.

Fact 8. Let $x > 3$, $y > 3$, N odd. For any compatible pair A, B there exists a h-path $h = A::B$ if $d(A,V)$ is even, where V is any vertex in R.

(N is odd, therefore if $d(A,V)$ is even $d(B,V)$ is also even). To prove fact 8 operate on R^*, noting that N odd implies x and y odd, hence x^* odd. If $x^*=3$, then apply fact 5 (interchanging x and y). If $x^*>3$, divide R^* in $n = \lfloor x^*/2 \rfloor$ rectangles $R1(3,y)$, $R2(2,y)$,...., $Rn(2,y)$, with A in R1 and B in Rn (figure 8). By applying fact 5 to R1, and fact 3 to R2,...,Rn, build a h-path on R^* as shown in figure 8.

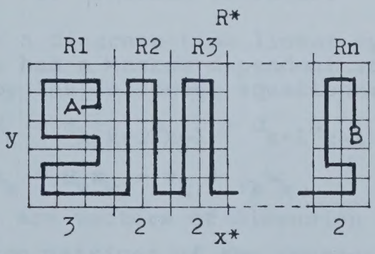


Fig. 8. h-path on R^*, for N odd and $d(A,V)$ even.

Fact 8 is an extension of fact 5, if part, for $x,y > 3$. For the only if part we do not have a reasonably clean proof, hence we formulate it as a conjecture.

Conjecture. If and only if holds in fact 8.

CONCLUDING REMARKS

We have shown how to build h-cycles and h-paths in any given rectangle with sides greater than one, and have discussed conditions for the existence of such paths. The proposed shapes for all paths evidently obey to simple construction rules, and can be easily described to be drawn by a computer program.

The existence conditions are immediate for h-cycles (fact 2), while several cases must be considered for h-paths. In all cases we have found necessary and sufficient conditions, except for $x, y > 3$ and N odd in fact 8, where the condition "d(A,V) is even" has been given as only sufficient. We have the proof that such a condition is also necessary [5]. However, our argument is so intrigued that we have preferred to formulate necessity as a conjecture, while looking for a more brilliant proof.

REFERENCES

[1] A.K.Chandra, H.Chang and C.K.Wong. 2-dimensional fast access to contiguous data in 2-dimensional magnetic bubble memories. U.S. Patent Pending.

[2] E.Lodi, F.Luccio, C.Mugnai, L.Pagli. On two dimensional data organization, I. Fundamenta Informaticae 2 (1978).

[3] D.T.Lee and C.K.Wong. Voronoi diagrams in L_1 (L_∞) metrics with 2-dimensional storage applications. IBM T.J.Watson Research Center.

[4] W.T.Tutte. Non-Hamiltonian planar maps. In: R.C.Read, Ed. Graph Theory and Computing. Academic Press (1972).

[5] F.Luccio and C.Mugnai. How to traverse a rectangle. Istituto di Scienze dell'Informazione, University of Pisa (1978).

SEQUENTIAL DETECTION WITH MARKOV INTERRUPTED OBSERVATIONS

M.T. HADIDI AND S.C. SCHWARTZ
Dept. of Electrical Engineering and Computer Science
Princeton University, Princeton, New Jersey 08540

ABSTRACT

We consider the sequential detection of a Markov sequence in a linear system with interrupted observations, i.e., systems with a switching environment. Because of the demanding memory requirements for optimum procedures, three sub-optimum filters are discussed, all of which feed into a sequential likelihood-ratio detector. The results of a computer simulation are presented.

I. INTRODUCTION

It is common to assume linear models for the state and observation when formulating dynamical system equations. This simplifies the analysis and, provided a suitable criterion is chosen for optimization, yields attractive solutions such as the familiar Kalman filter. There is, however, an underlying assumption in such a formulation, namely; the origin of observation is known. One example where this may not be true in practice is in target tracking, [1], where the sensor receives returns that could originate from one of various possible targets. In this case, the structure of the optimal solution may change completely as has been demonstrated in the switching environment problem analyzed by Ackerson and Fu, [2]. We shall address a problem related to this practical aspect of the model, and derive the sequential detector for a discrete-time linear system whose observation source switches in a Markovian manner.

In the next section, a statement of the problem under consideration is given and it is formulated using a minimum probability of error criterion. Regarding it as a hypothesis testing problem, we then apply Bayesian decision theory to get the optimal detector in Sec. III. Though the detector derived has a recursive structure, it requires numerical integration of, generally, non-Gaussian p.d.f.'s. Therefore, we also propose three suboptimal detectors in Sec. IV, and present comparison results based on simulation in Sec. V. Sec. VI contains comments and discusses possible extensions of the present work.

II. PROBLEM STATEMENT

We are given a discrete-time linear system in which the measurement noise has a Markov dependent statistical property. It is described by the following equations:

$$x_k = \phi_{k,k-1} x_{k-1} + G_{k-1} u_{k-1} \tag{1}$$

$$z_k = H_k x_k + v_k + \gamma_k w_k \tag{2}$$

where x_k and u_{k-1} are vectors of dimension $n \times 1$ and $r \times 1$ while $\phi_{k,k-1}$ and G_{k-1} are matrices of the appropriate dimension. We assume that the initial state x_0 is normally distributed and the sequence $\{u_k\}$ is white Gaussian with zero mean. Thus:

$$x_0 \sim N(\cdot|\mu_0,V_0) \ , \tag{3}$$

$$u_k \sim N(\cdot|0,V_u(k)) \ , \ E\{u_j u_k^T\} = V_u(k)\,\delta_{j,k} \ . \tag{4}$$

The state vector enters the measurement equation, (2), linearly and is corrupted by v_k or v_k+w_k depending on whether ν_k is 0 or 1, respectively. The vectors z_k, v_k and w_k are all of dimension m×1 and H_k is an m×n matrix. Again we assume $\{v_k\}$ and $\{w_k\}$ to be white noise sequences with the following statistics:

$$u_k \sim N(\cdot|0,V_v(k)) \ , \tag{5}$$

$$w_k \sim N(\cdot|0,V_w(k)) \ . \tag{6}$$

The sequence $\{\nu_k\}$ is a binary Markov chain defined on the state space $\{0,1\}$ and is statistically described by an initial probability vector $(1-p_0,p_0)^T$ and a transition probability matrix

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix} \ . \tag{7}$$

In the above it is assumed that the vector x_0 and the sequences $\{u_k\},\{v_k\}$ and $\{w_k\}$ as well as $\{\nu_k\}$ are all mutually independent.

Our problem is to decide - at each stage k - on the value of ν_k minimizing the probability of error in detection. Formally,

$$\begin{aligned} &\underset{\hat{\nu}_k}{\text{minimize}} && \Pr\{\hat{\nu}_k \neq \nu_k | z_0,\ldots,z_k\} \\ &\text{where} && \hat{\nu}_k = \hat{\nu}_k(z_0,\ldots,z_k) \\ & && z_k, x_k \text{ satisfy Eqs. (2) and (1) with the} \\ & && \text{underlying statistics given by Eqs.(3)-} \\ & && \text{(7).} \end{aligned} \tag{8}$$

The derivation of the detector is outlined in the following section.

III. THE BAYESIAN OPTIMAL DETECTOR

The minimum probability of error problem of Sec. II, is equivalent to a Bayesian decision problem in which we minimize the Bayes' risk for a special choice of the cost matrix (see for example [3]). Therefore, the problem under consideration reduces to testing - at each time k - the hypotheses:

$$\begin{aligned} &H_1\colon z_k = H_k x_k + v_k \\ \text{vrs.}\quad & \\ &H_2\colon z_k = H_k x_k + v_k + w_k \end{aligned} \tag{9}$$

where x_k evolves according to Eq. (1) and the true hypothesis switches from one stage to another according to the transition probability matrix P of Eq. (7).

By using Eq. (12), Ch. 2 of [3], the Bayesian optimal rule can be written as:

$$L_k(z^k) \overset{\Delta}{=} \frac{f(z^k|\gamma_k=1)}{f(z^k|\gamma_k=0)} \overset{H_2}{\underset{H_1}{\gtrless}} \frac{\text{prior probability } H_1 \text{ is true}}{\text{prior probability } H_2 \text{ is true}} \tag{10}$$

The explicit expression for the likelihood ratio, $L_k(z^k)$, can be obtained from the assumed Markov property and by applying Bayes' rule to $f(z^k|\gamma_k)$. The result is

$$L_k(z^k) = \frac{f(z_k|z^{k-1},\gamma_k=1)}{f(z_k|z^{k-1},\gamma_k=0)} \times$$

$$\frac{\left[1 + L_{k-1}(z^{k-1})\frac{P(1|1)}{P(1|0)}\frac{\Pr\{\gamma_{k-1}=1\}}{\Pr\{\gamma_{k-1}=0\}}\right] 1 + \frac{P(0|1)}{P(0|0)}\frac{\Pr\{\gamma_{k-1}=1\}}{\Pr\{\gamma_{k-1}=0\}}}{\left[1 + L_{k-1}(z^{k-1})\frac{P(0|1)}{P(0|0)}\frac{\Pr\{\gamma_{k-1}=1\}}{\Pr\{\gamma_{k-1}=0\}}\right] 1 + \frac{P(1|1)}{P(1|0)}\frac{\Pr\{\gamma_{k-1}=1\}}{\Pr\{\gamma_{k-1}=0\}}} \tag{11}$$

$$\overset{H_2}{\underset{H_1}{\gtrless}} \frac{\Pr\{\gamma_k=0\}}{\Pr\{\gamma_k=1\}}$$

Eq. (11) suggests that we can carry out our detection scheme in a sequential manner by using $L_{k-1}(z^{k-1})$, $\Pr\{\gamma_{k-1}=i\}, i = 0,1$ - obtained at stage $(k-1)$ - and by computing the density of z_k under each hypothesis conditioned on the previous observations. Another implication of Eq. (11) is that it reduces to Scharf's and Nolte's result, [4], if we set $P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ which they assumed.

To evaluate the p.d.f.'s and the probabilities appearing in Eq. (11), one uses the law of total probability, the Markov property and the system equations (1)-(2) to get:

$$p(\gamma_k) = p(\gamma_k|\gamma_{k-1}=0)\Pr\{\gamma_{k-1}=0\} + p(\gamma_k|\gamma_{k-1}=1)\Pr\{\gamma_{k-1}=1\} \tag{12}$$

$$f(z_k|z^{k-1},\gamma_k) = \int_{R^n} dx_k \cdot N(z_k|H_k x_k, V(k)) \cdot f(x_k|z^{k-1},\gamma_k) \tag{13}$$

$$f(x_k|z^{k-1},\gamma_k) = \sum_{i=0}^{1} p(\gamma_{k-1}=i|\gamma_k,z^{k-1}) \cdot f(x_k|z^{k-1},\gamma_{k-1}=i) \tag{14}$$

$$f(x_k|z^{k-1},\gamma_{k-1}=i) = \int_{R^n} dx_{k-1} \cdot N(x_k|\phi_{k,k-1}x_{k-1}, G_{k-1}V_u(k-1)G_{k-1}^T) \cdot \frac{f(z_{k-1}|x_{k-1},\gamma_{k-1}=i)\, f(x_{k-1}|z^{k-2},\gamma_{k-1}=i)}{f(z_{k-1}|z^{k-2},\gamma_{k-1}=i)} \tag{15}$$

where $V(k)$ is either $V_v(k)$ or $V_v(k) + V_w(k)$ depending on γ_k

being 0 or 1, respectively. The interested reader may refer to [5] for more details.

Eqs. (11)-(15) constitute the basis for a sequential procedure that decides on the true hypothesis or, equivalently, on γ_k.

IV. SUBOPTIMAL SEQUENTIAL DETECTORS

The optimal sequential procedure outlined in Section III is, at least in principle, straightforward to carry out. However, its implementation is not so simple. This stems from the requirement in Eqs. (13) and (15) to evaluate p.d.f.'s and perform numerical integrations.

One method to simplify the computation is by using a Gaussian approximation. More specifically, notice that if $f(x_k|z^{k-1},\gamma_k)$ in Eq. (13) were Gaussian, then it is completely specified by its mean and variance. Furthermore, under this assumption, $f(z_k|z^{k-1},\gamma_k)$ is also Gaussian with its mean and variance readily obtained through simple algebraic calculation; therefore avoiding the need for numerical integration. We make use of this fact and approximate $f(x_k|z^{k-1},\gamma_k)$, first by expressing it as

$$f(x_k|z^{k-1},\gamma_k) = \int_{R^n} dx_{k-1}\cdot N(x_k|\phi_{k,k-1}x_{k-1},G_{k-1}V_u(k-1)G_{k-1}^T) \cdot f(x_{k-1}|z^{k-1},\gamma_k) \ , \quad (16)$$

and then by setting $f(x_{k-1}|z^{k-1},\gamma_k=i)\doteq N(\cdot|\mu(k-1),V(k-1))$, for $i=0,1$. Here, $\mu(k-1)$ and $V(k-1)$ are chosen as the estimate of x_{k-1} given $(z_0,\ldots,z_{k-1})$ and the corresponding variance, which will clearly depend on the estimation method used.

The approximate p.d.f.'s can now be substituted into the optimal detector structure, Eqs. (11)-(13), yielding the required suboptimal sequential detectors.

Three estimators were used for evaluating $\mu(\cdot)$ and $V(\cdot)$. We shall briefly describe the underlying principle for each and give the resulting equations.

A. Decision-directed filtering

This estimation procedure assumes that the decisions we make about γ_k are correct. Consequently, the origin of the observations is no longer unknown and the conditional density $f(x_k|z^{k-1},\gamma_k)$ may be written as

$$f(x_k|z^{k-1},\gamma_k) \doteq f(x_k|z^{k-1},\hat{\Gamma}_{k-1},\gamma_k) = N(x_k|x^{(1)}_{k|k-1},V^{(1)}(k|k-1)) \quad (17)$$

where $\hat{\Gamma}_{k-1}=(\hat{\gamma}_0,\ldots,\hat{\gamma}_j,\ldots,\hat{\gamma}_k)$ and $\hat{\gamma}_j$ denotes the decision at time j. Observe that $x^{(1)}_{k|k-1}$ and $V^{(1)}(k|k-1)$ can be computed using the Kalman filter and hence are determined recursively, [6]. Upon using Eq. (17) together with Eqs. (11)-(13), we obtain the desired sequential detector which will be denoted by D-D.

B. Linear least-mean-squared error filtering

The estimate of x_k given the observations $(z_0,\ldots,z_k)$ may also be determined by considering all linear functions on z^k and choosing as the estimator the one with minimum error variance. The derivation uses the orthogonality principle and yields the following solution, [5]:

$$x^{(2)}_{k|k} = \phi_{k|k-1}x^{(2)}_{k-1|k-1} + K^{(2)}(k)[z_k - H_k\phi_{k,k-1}x^{(2)}_{k-1|k-1}] \tag{18}$$

$$V^{(2)}(k) = [I - K^{(2)}(k)H_k]V^{(2)}(k|k-1) \tag{19}$$

where

$$K^{(2)}(k)=V^{(2)}(k|k-1)H_k^T[H_kV^{(2)}(k|k-1)H_k^T+V_v(k)+\Pr\{\gamma_k=1\}V_w(k)]^{-1} \tag{20}$$

$$V^{(2)}(k|k-1)=\phi_{k,k-1}V^{(2)}(k)\phi^T_{k,k-1}+G_{k-1}V_u(k-1)G^T_{k-1},\; V^{(2)}(0)=V_0 \tag{21}$$

Notice that the estimator structure is decision independent, and that only the first-order statistics of γ_k are used. It is expected, therefore, that this filter will perform better than the D-D filter and by incorporating higher order statistics of γ_k into the filter structure one may obtain even a better performance. This leads to the next estimation procedure.

C. Least-mean-squared-error nonlinear filtering (approximate non-Gaussian filtering)

A logical approach to improve the estimate of x_k given z^k is to relax the linear constraint, consider all functions on z^k and select the one with the minimum error variance. This is the idea of the approximate non-Gaussian filter which got its name because it incorrectly assumes the prior p.d.f., $f(x_k|z^{k-1})$, to be Gaussian. By extending the results derived by Masreliez, [7], to the case of Markov interrupted observations, we obtain the following estimate and its variance:

$$x^{(3)}_{k|k} = \phi_{k,k-1}x^{(3)}_{k-1|k-1} + V^{(3)}(k|k-1)H_k^T g(z_k)\ , \tag{22}$$

$$V^{(3)}(k) = [I - V^{(3)}(k|k-1)H_k^T G(z_k)H_k]V^{(3)}(k|k-1)\ , \tag{23}$$

where the expressions for g(•) and G(•) are derived in [5]. There, it is shown that both g and G are nonlinear functions of z_k through the posterior probability $p(\gamma_k|z^k)$. Since this dependency incorporates the high-order statistics of γ_k into the filter, one may expect to get better performance than the linear LMSE filter, especially if the actual $f(x_k|z^{k-1})$ is close to Gaussian. This turned out to be the case for the system simulated, as discussed in the next section.

V. SIMULATION RESULTS

We performed a simulation study on a 1-dimensional system governed by the following equations:

$$x_k = -.8x_{k-1} + u_{k-1}$$

$$z_k = x_k + v_k + \gamma_k w_k$$

with x_0, u_k, v_k all $\sim N(\cdot|0,1)$, $w_k \sim N(\cdot|0,V_w)$, and $\gamma_k \epsilon \{0,1\}$ with transition matrix $P = \begin{bmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{bmatrix}$.

For fixed values of V_w and α, γ_k was detected using the three suboptimal detectors; each corresponding to one of the state estimation procedures outlined in Sec. IV. Performance was compared in terms of the mean-squared-error in estimation (MSE), and the probability of error in detection (both were evaluated from 3000 Monte Carlo runs.) By varying the values of V_w and α we also changed the observation noise distribution and studied its effect on each detector, thereby examining the detector sensitivity. The results obtained are shown in Figs. 1,2 and 3.

In Fig. 1, the noise component w has variance V_w=10 and the sequence $\{\gamma_k\}$ is i.i.d. with $Pr\{\gamma_k=1\}=0.5$. It is seen that the MSE of the approximate non-Gaussian procedure is uniformly less than that of the linear LMSE detector and the latter outperforms the D-D procedure. Thus the results confirm our remarks in Sec. IV. On the other hand, Fig. 1-b shows the detection performance of the three procedures to be almost the same, a result that does not have an obvious explanation but lends support to a similar conclusion reported by Davis and Andreadakis in [8]. These general features of the estimation and detection performances occured again in Fig. 2 which corresponds to V_w=10 and α=0.9 (dependent switching sequence.)

Fig. 3 depicts the sensitivity results for the approximate non-Gaussian procedure. Here we observe that both the MSE and error probability increased when the switching sequence was no longer independent (α=0.9 and 0.99). Furthermore, by reducing V_w to 1 and fixing α at 0.5, the MSE dropped while the error probability increased. This behavior is expected; for $x_{k|k}$ is a function of $z_0,\ldots,z_k$ and the uncertainty of the observations decreases as V_w is reduced, thereby reducing the mean squared error of $x_{k|k}$. On the otherhand, V_w can be regarded as the signal power in the detection of γ_k and, consequently, as its value is reduced so is the signal-to-noise ratio and therefore

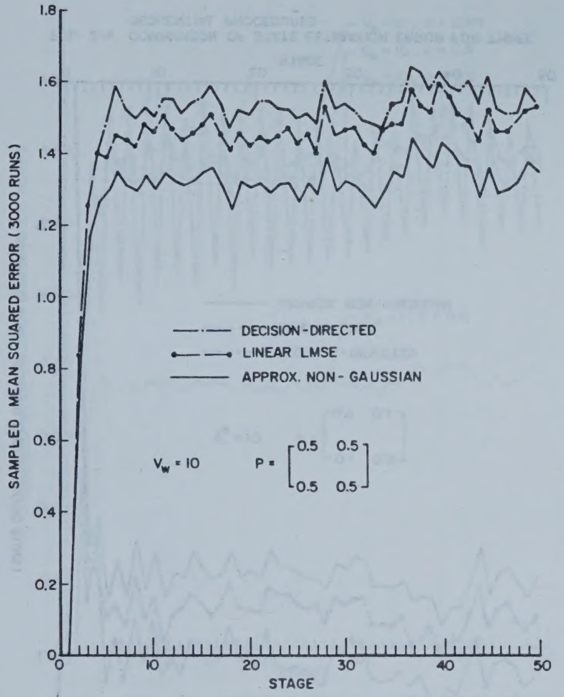


FIG. 1-a COMPARISON OF STATE ESTIMATION ERROR FOR THREE SEQUENTIAL PROCEDURES

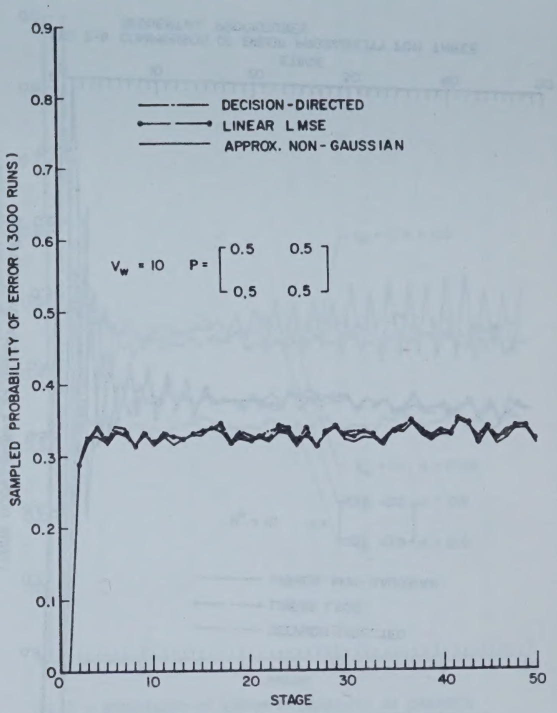


FIG. 1-b COMPARISON OF ERROR PROBABILITY FOR THREE SEQUENTIAL PROCEDURES

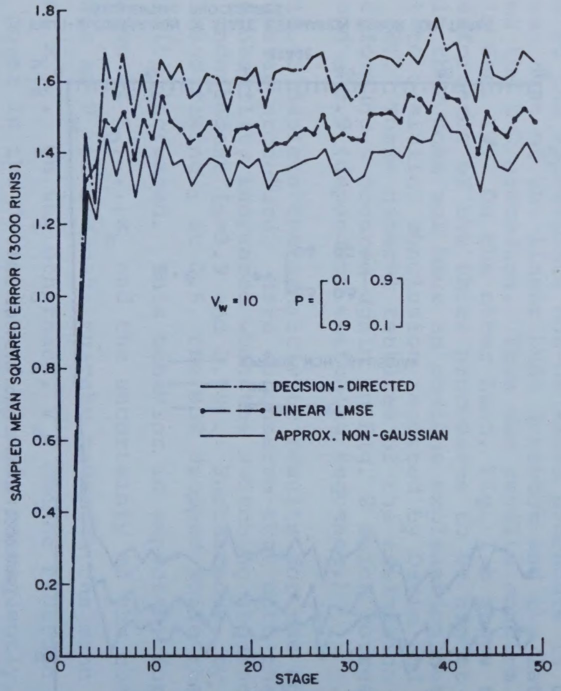


FIG. 2-a COMPARISON OF STATE ESTIMATION ERROR FOR THREE SEQUENTIAL PROCEDURES

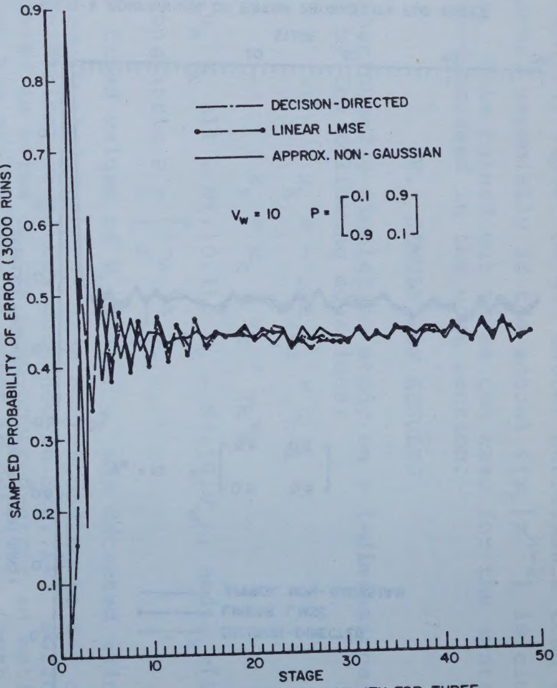


FIG. 2-b COMPARISON OF ERROR PROBABILITY FOR THREE SEQUENTIAL PROCEDURES

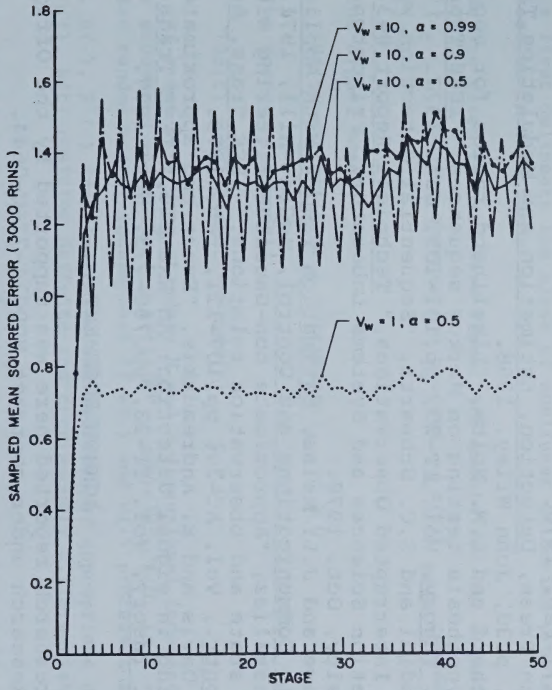


FIG. 3-a SENSITIVITY OF STATE ESTIMATION ERROR TO CHANGES IN NOISE VARIANCE, V_w, AND TRANSITION MATRIX, P. (APPROX. NON-GAUSSIAN PROCEDURE)

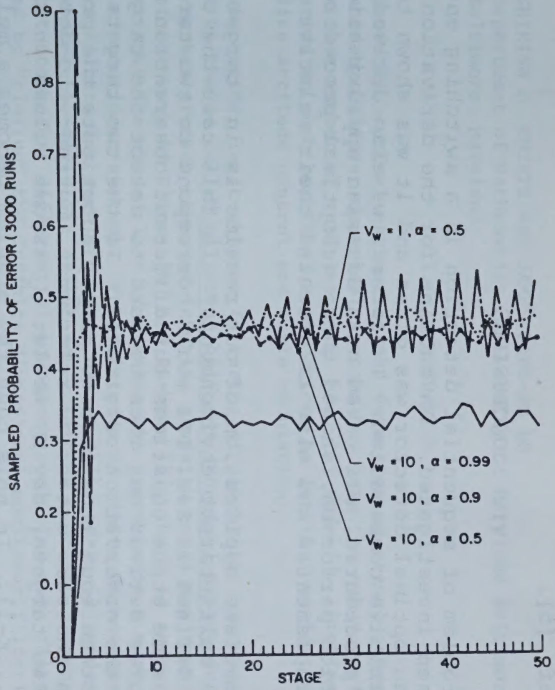


FIG. 3-b SENSITIVITY OF ERROR PROBABILITY TO CHANGES IN NOISE VARIANCE, V_w, AND TRANSITION MATRIX, P. (APPROX. NON-GAUSSIAN PROCEDURE)

the error probability increases. Other sensitivity results are given in ref. [5].

VI. CONCLUSIONS

The problem of sequential detection in a switching environment has been investigated. An outline for the derivation of the Bayesian optimal detector was given and it was shown to require prohibitive computation. Therefore, we also derived three suboptimal procedures, all based on a Gaussian approximation. The simulation performed, showed these suboptimal procedures to be easily implemented and also illustrated their relative performance.

An immediate application of our results is in "target tracking in multi-target environment." In this case the two hypotheses defined in Sec. III would correspond to two targets having the same state equations but different observation models. The procedures derived can then be used to detect the target from which the observation originated. If the two targets have different state equations then we simply substitute the proper p.d.f.'s in the same detection relations. Further, to detect more than two targets, say K, we let γ_k assume values in the set $\{0,1,\ldots,K-1\}$. In this case the problem becomes a multiple hypothesis testing with slightly more complicated expressions.

REFERENCES

[1] R.G. Sea, "An efficient suboptimal decision procedure for associating sensor data with stored tracks in real-time surveillance systems," in Proc. 1971 IEEE Conf. on Decision and Control, pp.33-37, Miami Beach, Fla., Dec. 1971.

[2] G.A. Ackerson and K.S. Fu, "On state estimation in switching environments," IEEE Trans. Automat. Contr., Vol. AC-15, pp.10-17, Feb. 1970.

[3] H.L. Van Trees, Detection, Estimation and Modulation Theory, Part I, p.30, John Wiley, 1968.

[4] L.L. Scharf and L.W. Nolte, "Likelihood ratios for sequential hypothesis testing on Markov sequences," IEEE Trans. Inform. Theory, Vol. IT-23, pp.101-109, Jan. 1977.

[5] M.T. Hadidi and S.C. Schwartz, "Sequential detection with Markov Interrupted Observations," Technical Report #43, Information Sciences and Systems Laboratory, Princeton University, Oct. 1978.

[6] A.P. Sage and J.L. Melsa, Estimation Theory With Applications to Communications and Control, McGraw-Hill, 1971.

[7] C.J. Masreliez, "Approximate non-Gaussian-filtering with linear state and observation relations," IEEE Trans. Automat. Contr., Vol. AC-20, pp.107-111, Feb. 1975.

[8] M.H.A. Davis and E. Andreadakis, "Exact and approximate filtering in signal detection: An Example," IEEE Trans. Inform. Theory, Vol. IT-23, pp.768-773, Nov. 1977.

ACKNOWLEDGEMENT

The research reported here was supported by the Office of Naval Research under contract N00014-77-C-0644.

OPTIMAL FILTERS FOR NILPOTENT ASSOCIATE-ALGEBRAIC BILINEAR SYSTEMS

SHIRISH D. CHIKTE and JAMES TING-HO LO
Department of Mathematics, University of Maryland Baltimore County
Baltimore, Maryland

1. INTRODUCTION

Recently, optimal estimation and detection of signal processes generated by bilinear dynamical systems has been the subject of investigation in a number of articles, e.g. [1], [2] and [3]. Problems of this type arise in such practical applications as inertial navigation, satellite attitude control and angle modulation.

We consider least-squares filtering of stochastic processes generated by the so called nilpotent bilinear dynamical system driven by a Gauss-Markov process observed in white gaussian noise environment. In view of the work by Fliess [4] and Sussman [5], there is strong reason to suspect that the above class of systems can approximate a much wider class of nonlinear systems. The existence of finite dimensional, recursive filters for such processes was established in [3] under a somewhat weaker Lie-algebraic nilpotency condition. In the present paper, explicit stochastic differential equations realizing such filters are derived by "closing" the infinite dimensional nonlinear filtering equations of Kushner [6]. It is found that the filter structure is not only bilinear but it also inherits the nilpotency property from the original system.

The next section presents the problem formulation. The third section contains the derivation of filter equations. The final section concludes with an example and computational considerations.

2. PROBLEM STATEMENT

We first delineate the class of nonlinear filtering problems addressed to in this paper.

We are given the standard linear Itô models for the signal process $\{\xi(t)\}_{t \geq 0}$ and the observation process $\{z(t)\}_{t \geq 0}$ respectively, as follows.

The signal model:

$$d\xi(t) = F(t)\xi(t)dt + Q^{\frac{1}{2}}(t)dw(t), \quad t \geq 0 \tag{2.1}$$

The observation model:

$$dz(t) = H(t)\xi(t)dt + R^{\frac{1}{2}}(t)dv(t), \quad t \geq 0 \tag{2.2}$$

where $w(\cdot)$ and $v(\cdot)$ are standard N and P dimensional independent Wiener processes respectively, $\xi(t) \in \mathbb{R}^N$, $z(t) \in \mathbb{R}^P$, $\xi(0)$ is a zero mean gaussian random vector independent of $w(\cdot)$ and $v(\cdot)$ processes and $F(\cdot)$, $Q^{\frac{1}{2}}(\cdot)$, $H(\cdot)$, $R^{\frac{1}{2}}(\cdot)$ are time-dependent matrices of appropriate dimensions with $Q(t)$, $R(t)$ positive definite and continuously differentiable for all t.

Now consider a process $\{x(t)\}_{t \geq 0}$ generated by a bilinear dynamical system of the following form:

$$dx(t) = \left(A + \sum_{i=1}^{N} B_i \xi_i(t)\right)x(t)dt, \quad x(t)\in \mathbb{R}^M, \; t \geqq 0 \qquad (2.3)$$

where $x(0)$ is a gaussian random vector independent of $\xi(0)$, $\{w(t)\}_{t \geqq 0}$, $\{v(t)\}_{t \geqq 0}$.

We are interested in an exact, finite dimensional system of stochastic differential equations for the least squares estimator $E[x(t)|z^t] \triangleq \hat{x}(t|t)$ of $x(t)$. It is well-known (see e.g. [6]) that this, in general, is not possible. In this paper, we intend to derive such a filter and study its characteristics with the proviso that the matrices A and $\{B_i\}_{i=1}^{N}$ in (2.3) satisfy a well-known algebraic property called nilpotency.

Definition 1:

An associative algebra $\mathcal{A}$ of matrices is said to be nilpotent (of order K) if there exists a positive integer K such that the (matrix) product of any K or more matrices in $\mathcal{A}$ vanishes.

Assumption 2:

The associative algebra $\mathcal{A}$ generated by the set of matrices $\{Ad_A^k(B_i) \,|\, i=1,2,\ldots,N;\ k=0,1,\ldots\}$ is nilpotent (of order K, say) where

$$Ad_A^o(B_i) \triangleq B_i$$

and

$$Ad_A^k(B_i) \triangleq [A, Ad_A^{k-1}(B_i)] \triangleq A\cdot Ad_A^{k-1}(B_i) - Ad_A^{k-1}(B_i)\cdot A$$

for $k = 1,2,\ldots$

Remark:

Assumption 2 is stronger than the Lie-algebraic nilpotency assumption made in [3].

3. THE STRUCTURE OF THE NONLINEAR FILTER

The filtering problem posed in the last section, was shown to be amenable to solution in [3]. In the main theorem of this paper that follows shortly, we explicitly exhibit the structure of the nonlinear filter and study its characteristics. We begin with some lemmas.

Lemma 3:

The input-output map realized by the system (2.3) satisfying Assumption 2 can also be realized by a bilinear system which is a "direct sum" of (a finite number of) finite dimensional systems of the following form

$$d\bar{x}^k(t) = \left(\bar{A}^k + \sum_{i=1}^{N} \bar{B}_i^k \xi_i(t)\right) \bar{x}^k(t)dt, \quad t \geqq 0 \qquad (3.1)$$

where,

$$\bar{A}^k = \begin{bmatrix} A_1^k & 0 & \cdots & \cdots & 0 \\ 0 & A_2^k & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & A_{M_k}^k \end{bmatrix} \qquad (3.2)$$

with A_i^k, $i=1, \ldots, M_k$ being $J_i \times J_i$ Jordan blocks, and

$$\bar{B}_i^k = \begin{bmatrix} 0 & B_{i,1}^k & 0 & \cdots & \cdots & 0 \\ \vdots & 0 & B_{i,2}^k & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ \vdots & & & & 0 & B_{i,M_k-1}^k \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} \qquad (3.3)$$

with $B_{i,j}^k$, $i=1,\ldots,N$; $j=1,\ldots,M_k-1$ being $J_j \times J_{j+1}$ matrices.
Hence, the associative algebra $\left\{ Ad_{\bar{A}}^k (\bar{B}_i) \middle| k=0,1,\ldots;\ i=1,\ldots,N \right\}_{AA}$ is also nilpotent of the same order K.

Proof:

The proof follows closely the ideas in the proofs of theorems 4 and 6 of Brockett [7] and hence is kept brief.

Using the transformation $z(t) = e^{-At} x(t)$ and applying the Peano-Baker formula, we find that the Volterra series solution to (2.3) is given by

$$x(t) = \Big[e^{At} + \int_0^t e^{A(t-\sigma_1)} \Big(\sum_{i=1}^N B_i \xi_i(\sigma_1) \Big) e^{A\sigma_1} d\sigma_1$$

$$+ \int_0^t \int_0^{\sigma_1} e^{A(t-\sigma_1)} \Big(\sum_{i=1}^N B_i \xi_i(\sigma_1) \Big) \cdot e^{A(\sigma_1-\sigma_2)}$$

$$\Big(\sum_{j=1}^N B_j \xi_j(\sigma_2) \Big) e^{A\sigma_2} d\sigma_1 d\sigma_2 + \ldots\ldots. \Big] \; x(0) . \qquad (3.4)$$

It now follows from the Baker-Campbell-Hausdorff formula together with Assumption 2 that all the terms in (3.4) beyond the first K terms will vanish.

Now, let the Jordan form of A be given by $\mathrm{diag}\left\{ A_1, A_2, \ldots, A_s \right\}$ where A_i is, say, $J_i \times J_i$. Let $K_i \triangleq \sum_{k=1}^{i-1} J_k$, $i=1,\ldots,s$ with

$K_1 \triangleq 0$, and define $y^i \triangleq [x_{K_i+1} \cdots\cdots x_{K_{i+1}}]^T$, $i=1,\ldots,s$. Recall that e^{At} consists of $J_i X J_k$ blocks, $i,k=1,\ldots,s$ which can be written as $\sum_{\ell=1}^{s} C_\ell e^{A_\ell t} D_\ell$, for some $J_i X J_\ell$ matrices C_ℓ and $J_\ell X J_k$ matrices D_ℓ. With these remarks in mind, it is not difficult to see that the r^{th} order term, $1 \leqq r < K$, in the Volterra series (3.4), for the subset $y^i(\sigma_o)$ of components of $x(\sigma_o)$, $i=1,\ldots,s$, consists of groups of terms of the form

$$\int_o^{\sigma_o}\int_o^{\sigma_1}\cdots\int_o^{\sigma_{r-1}} \sum_{m_1,\ldots,m_r=1}^{N} C_1 e^{A_1(\sigma_o-\sigma_1)} D_1 e^{A_2(\sigma_1-\sigma_2)} D_2 \cdots D_r e^{A_r\sigma_r} y^k(0)$$

$$\prod_{i=1}^{r} \xi_{m_i}(\sigma_i)d\sigma_i$$

But each group of this type is realizable by a system of the type claimed in the lemma with $\bar{A}$ and $\left\{\bar{B}_i\right\}_{i=1}^{N}$ as in (3.2), (3.3) (see theorem 4 of [7] for details). The final assertion regarding nilpotency is obvious if we note that $\bar{A}$ is upper triangular while $\bar{B}_i$, $i=1,\ldots,N$ are strictly upper triangular.

Lemma 4:

Consider the signal and observation models of (2.1) and (2.2) respectively. Define a (vector) process $\left\{y(t)\right\}_{t \geqq 0}$ by

$$dy(t) = Dy(t)dt + \sum_{i=1}^{N} E_i \xi_i(t) y(t)$$

where $D, \left\{E_i\right\}_{i=1}^{N}$ are matrices of appropriate dimension, $y(0)$ is independent of $\xi(0)$, $w(\cdot)$ and $v(\cdot)$ processes. (3.5)

Then $\hat{y}(t|t) \triangleq E[y(t)/z^t]$ satisfies the following stochastic differential equation:

$$d\hat{y}(t|t) = D\hat{y}(t|t)dt + \sum_{i=1}^{N} E_i \cdot E^t[\xi_i(t)y(t)]dt$$

$$+ \left\{E^t[y(t)\xi^T(t)] - \hat{y}(t|t)\hat{\xi}^T(t|t)\right\} H^T(t)R^{-1}(t)d\nu(t)$$

$$\hat{y}(0|0) = E[y(0)] \tag{3.6}$$

where,

$$E^t[\cdot] \triangleq E[\cdot|z^t] \triangleq E[\cdot|\left\{z(\tau) \mid 0 \leqq \tau \leqq t\right\}]$$

$$d\nu(t) = dz(t) - H(t)\hat{\xi}(t|t)dt .$$

Proof:

Apply the Kushner nonlinear filtering equations [6] to the signal process $(y^T(\cdot),\xi^T(\cdot))^T$ with $z(\cdot)$ as the observation process.

Lemma 5:

Let $x = (x_0,x_1,\dots,x_n)^T$ be a gaussian random vector with mean vector $m = (m_0,m_1,\dots,m_n)^T$ and covariance matrix $P = [P_{ij}]_{i\cdot j=0}^{n}$. We then have the following relation:

$$E[\prod_{i=0}^{n} x_i] = \begin{cases} m_0 E[\prod_{i=1}^{n} x_i] + \sum_{j=1}^{n} P_{0j} E[\prod_{\substack{i=2\\ i\neq j}}^{n} x_i], & n > 1 \\ \\ m_0 m_1 + P_{01}, & n = 1. \end{cases} \tag{3.7}$$

Proof: See e.g. [8].

We are now ready to state and prove the main theorem of this paper.

Theorem 6:

Consider the signal process $\{x(t),\xi(t)\}_{t\geq 0}$ evolving as in (2.3) and (2.1) and the observation process $\{z(t)\}_{t\geq 0}$ as in (2.2). Assume that (2.3) satisfies Assumption 2. Then the least squares filtered estimate $\hat{x}(t|t)$ can be obtained via the following finite dimensional system of bilinear stochastic differential equations with suitable initial conditions:

$$d\hat{x}^*(t|t) = (A^*(t)dt + \sum_{i=1}^{N} B_i^* \hat{\xi}_i(t|t)dt + \sum_{i=1}^{N} C_i^* d\mu_i(t))\hat{x}^*(t|t)dt$$

$$\hat{x}(t|t) = L\hat{x}^*(t|t)\,, \quad t \geq 0 \tag{3.8}$$

where $\hat{\xi}(t|t)$, $t \geq 0$ is given by the standard Kalman-Bucy filter [9], $A^*(\cdot)$ is a deterministic matrix time function. $L, \{B_i^*, C_i^*\}_{i=1}^{N}$ are constant matrices, and

$$\mu(t) \triangleq [\mu_1(t)\ \mu_2(t) \cdots \mu_N(t)]^T \triangleq \int_0^t H^T(\tau)\, R^{-1}(\tau) d\nu(\tau)$$

is the modified innovations process.

Furthermore, the associative algebra generated by the following set of matrices,

$$\left\{ \mathrm{Ad}^k_{(\mathrm{Ad}^{\ell}_{C_i^*}(A^*(t)))}(B_j^*) \,\middle|\, i,j=1,2,\dots,N; k,\ell=0,1,2,\dots \right\} \quad t \geq 0$$

is nilpotent of order K.

Proof:

From Lemma 3, it is clear that we may take - without loss of generality - (2.3) to be a stationary bilinear system in which the matrices A and $\{B_i\}_{i=1}^{N}$ are of the special form stipulated in (3.2) and (3.3) respectively. Furthermore, we shall assume for ease of exposition, that

$A = \operatorname{diag}(a_1, a_2, \ldots, a_M)$; consequently $B_k \triangleq [b_k^{i,j}]$ is such that

$$b_k^{i,j} = \begin{cases} 0 \ , & j \neq i+1 \\ b_2^k \ , & j = i+1 \end{cases} \ , \ k=1,\ldots,N \ \text{ and } \ x_i(0) = 0 \ \text{ a.s. } \ \text{for } i < M.$$

For, it would become apparent in the sequel, the proof for the general case would involve little more than additional notational complexity.

We now apply Lemma 4 to (2.3) with the above mentioned specialization. With the help of Lemma 5, the resulting differential equation analogous to (3.6) can be written in the following form:

$$d\hat{x}(t|t) = (A + \sum_{j=1}^{N} B_j \hat{\xi}_j(t|t))\hat{x}(t|t)dt + \sum_{j=1}^{N} B_j \hat{x}^j(t|t)dt$$
$$+ [\hat{x}^1(t|t), \hat{x}^2(t|t), \ldots, \hat{x}^N(t|t)]H^T(t)R^{-1}(t)d\nu(t) \tag{3.9}$$

where the "augmenting states" appearing in (2.17) are defined as

$$x^j(\cdot) \triangleq [x_1^j(\cdot), x_2^j(\cdot), \ldots, x_M^j(\cdot)]^T \ , \ j=1,2,\ldots,N \ ,$$

$$x_q^k(\sigma_0) = \begin{cases} \displaystyle\sum_{m_1,\ldots,m_{M-q}=1}^{N} \ \sum_{r=1}^{M-q} \int_o^{\sigma_0}\int_o^{\sigma_1}\cdots\int_o^{\sigma_{M-q-1}} P_{k,m_r}(\sigma_0,\sigma_r)e^{a_{q+r}(\sigma_r-\sigma_{r+1})} \cdot \\ \displaystyle\cdot b_{q+r}^{m_r} \cdot e^{a_M\sigma_{M-q}} x_M(0) \Big(\prod_{\substack{i=0 \\ i\neq r}}^{M-q-1} e^{a_{q+i}(\sigma_i-\sigma_{i+1})} {}_{m_{i+1}} \cdot b_{q+i}^{m_{i+1}} \\ \cdot \xi_{m_{i+1}}(\sigma_{i+1})d\sigma, \quad q < M \\ 0, \qquad q=M \end{cases} \tag{3.10}$$

(Note: In the general case $x_q^k(\sigma_0)$ would be a vector of a dimension consistent with the q^{th} Jordan block).

and,

$$P_{i,j}(t,\sigma), \quad i,j=1,\ldots,N \ ; \ 0 \leq \sigma \leq t \ ,$$

is the i,j^{th} element of the conditional covariance matrix

$$P(t,\sigma) = E[(\xi(t) - \hat{\xi}(t|t))(\xi(\sigma) - \hat{\xi}(\sigma|t))^T \Big| z^t \,] \tag{3.11}$$

Also recall [10] that $P(t,\sigma)$, $\sigma \leq t$ satisfies the following differential equation

$$\frac{\partial P(t,\sigma)}{\partial t} = [F(t) - P(t)H^T(t)R^{-1}(t)H(t)]P(t,\sigma) \ . \tag{3.12}$$

Now, if we differentiate the augmenting state vector

$${}^1x(\cdot) \triangleq [x^{1T}(\cdot) \ x^{2T}(\cdot), \ \ldots \ x^{NT}(\cdot)]^T \in \mathbb{R}^{MN} \ , \tag{3.13}$$

then utilizing (3.12), we find that

$$d\,{}^{1}x(t) = \alpha^{1}(t)\,{}^{1}x(t)dt + \beta^{1}(t)x(t)dt + \sum_{i=1}^{N} \gamma_i^{1}\,{}^{1}x(t)\,\xi_i(t)dt \qquad (3.14)$$

where $\alpha^1(\cdot)$, $\beta^1(\cdot)$, $\left\{\gamma_i^1\right\}_{i=1}^N$ are determined by the problem parameters and the error covariance $P(\cdot)$, the $(M \times M)$ blocks of $\alpha^1(\cdot)$ and $\beta^1(\cdot)$ are upper triangular and $\left\{\gamma_i^1\right\}_{i=1}^N$ are block diagonal matrices with identical strictly upper triangular blocks given by

$$\begin{bmatrix} 0 & b_1^i & 0 & \cdots & \cdots & 0 \\ 0 & 0 & b_2^i & 0 & & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & b_{M-2}^i & 0 \\ \vdots & & & & \ddots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 0 \end{bmatrix} \triangleq \text{a block of } \gamma_i^1 \qquad (3.15)$$

Now we may again apply Lemmas 4 and 5 to (3.14), and get

$$\begin{aligned} d\,{}^{1}\hat{x}(t|t) &= \left(\alpha^{1}(t) + \sum_{i=1}^{N} \gamma_i^{1}\,\hat{\xi}_i(t|t)\right){}^{1}\hat{x}(t|t)dt + \beta^{1}(t)\hat{x}(t|t)dt \\ &+ \sum_{j=1}^{N} \gamma_j^{1}\,{}^{1}\hat{x}^{j}(t|t)dt + \left[{}^{1}\hat{x}^{1}(t|t),\ {}^{1}\hat{x}^{2}(t|t), \ldots, {}^{1}\hat{x}^{N}(t|t)\right]H^{T}(t)R^{-1}(t)d\nu(t) \end{aligned} \qquad (3.16)$$

where the "new" set of augmenting states are

$${}^{1}x^{j}(\cdot) \triangleq \left[x^{j,1^{T}}(\cdot), x^{j,2^{T}}(\cdot), \ldots\ x^{j,N^{T}}(\cdot)\right]^{T}, \quad j=1,\ldots,N$$

$$x^{j,k}(\cdot) \triangleq \left[x_1^{j,k}(\cdot),\ x_2^{j,k}(\cdot), \ldots, x_M^{j,k}(\cdot)\right]^{T}$$

$$x_q^{j,k}(\sigma_o) = \begin{cases} \displaystyle\sum_{m_1,\ldots,m_{M-q}=1}^{N}\ \sum_{\substack{r_1,r_2=1 \\ r_1 \neq r_2}}^{M-q} \int_0^{\sigma_o}\int_0^{\sigma_1}\cdots\int_0^{\sigma_{M-q-1}} P_{j,m_{r_1}}(\sigma_o,\sigma_{r_1})P_{k,m_{r_2}}\cdot(\sigma_o,\sigma_{r_2}) \\ \quad e^{a_{q+r_1}(\sigma_{r_1}-\sigma_{r_1+1})}\, b_{q+r_1}^{m_{r_1}}\, e^{a_{q+r_2}(\sigma_{r_2}-\sigma_{r_2+1})}\cdot \\ \quad e^{a_M\sigma_{M-q}}\, x_M(0)\left(\displaystyle\prod_{\substack{i=0 \\ i\neq r_1,r_2}}^{M-q-1} e^{a_{q+i}(\sigma_i-\sigma_{i+1})}\cdot b_{q+i}^{m_{i+1}}\,\xi_{m_{i+1}}(\sigma_{i+1})d\sigma_{i+1}\right), \\ \quad q < M-1 \\ 0 \qquad q \geq M-1 \end{cases} \qquad (3.17)$$

Now the new augmenting state vector

$$ {}^{2}x(\cdot) \triangleq [{}^{1}x^{1}(\cdot)^{T}, {}^{1}x^{2}(\cdot)^{T}, \ldots, {}^{1}x^{N^{T}}]^{T} \in \mathbb{R}^{MN^{2}} \tag{3.18} $$

analogous to (3.16), is seen to satisfy the differential equation

$$ d\,{}^{2}x(t) = \alpha^{2}(t)\,{}^{2}x(t)d + \beta^{2}(t)\,{}^{1}x(t)dt + \sum_{i=1}^{N} \gamma_{i}^{2}\,{}^{2}x(t)\xi_{i}(t)dt \tag{3.19} $$

in which $M \times M$ blocks of $\alpha^{2}(\cdot)$, $\beta^{2}(\cdot)$ and γ_{i}^{2} have the same properties as those of $\alpha^{1}(\cdot)$, $\beta^{1}(\cdot)$ and $\gamma_{i}^{2}(\cdot)$ respectively with a block of γ_{i}^{2} obtained by setting $b_{M-2}^{i} = 0$ in the block (3.15) of γ_{i}^{1}. The above process can now be iterated and it is clear that this sequence will terminate at the M^{th} application - in the general case, this would be the total number of Jordan blocks - of Kushner equations since $\gamma_{i}^{M-1} = 0$, $i=1,2,\ldots,N$. Collecting together the differential equations for $x(\cdot), {}^{1}x(\cdot), \ldots {}^{M-1}x(\cdot)$, we see that the dimension of the resulting nonlinear filter equals $(M+MN+MN^{2}+\ldots+MN)^{M-1} = \frac{M}{N-1}(N^{M}-1)$. The filter form (3.8) follows upon writing the innovations term in the standard bilinear format, which also reveals the fact that the matrices $\left\{C_{i}^{*}\right\}_{i=1}^{N}$ consist of upper triangular $M \times M$ blocks. We thus see that $\left\{B_{i}^{*}\right\}_{i=1}^{N}$ consists of (strictly) upper triangular $(M \times M)$ blocks, while $\left\{A^{*}(t)\right\}_{t \geqq 0}$ and $\left\{C_{i}^{*}\right\}_{i=1}^{N}$ consist of $(M \times M)$ blocks that are upper triangular. It is, therefore, easy to see - if we carry out matrix multiplications blockwise - that the set of matrices $\left\{\mathrm{Ad}_{C_{i}^{*}}^{\ell}(A^{*}(t)) \,\middle|\, i \geqq 0; i=1,\ldots,N; \ell=0,1,\ldots\right\}$ also have $(M \times M)$ blocks all upper triangular. Consequently the set

$$ \left\{\mathrm{Ad}^{k}_{(\mathrm{Ad}^{\ell}_{C_{i}^{*}}(A^{*}(t)))}(B_{j}^{*}) \,\middle|\, t \geqq 0; i,j=1,\ldots,N; k,\ell=0,1,2,\ldots\right\} $$

has $(M \times M)$ blocks that are all strictly upper triangular. Hence, blockwise multiplication shows that the associative algebra generated by them is nilpotent of order K.

Q.E.D.

The filter structure revealed in the above theorem has some interesting features worth noting. We first see that analogous to the case of linear signal models, the "drift" terms in (3.8) do preserve the bilinear structure of the signal model (2.14) and also inherit its nilpotency property; nevertheless the matrix $A^{*}(\cdot)$ is time varying although the matrix A is constant. Secondly, the filter is bilinear in the innovations as well. Also observe that the state spaces of both the signal model and the filter are (not necessarily identical) nilpotent group manifolds. This suggests that it may be worthwhile to investigate optimal filters under criteria that are defined on such manifolds. Such an approach for abelian groups was followed in [1].

4. COMPUTATIONAL CONSIDERATIONS

Realization of the filter (2.16) in the form of a block schematic is shown in figure 1 on the following page. The practical importance of the bilinear property of the above filter is that a real-time analog implementation of the filter is still possible with easily available and cheap hardware consisting of integrators, summers and multipliers.

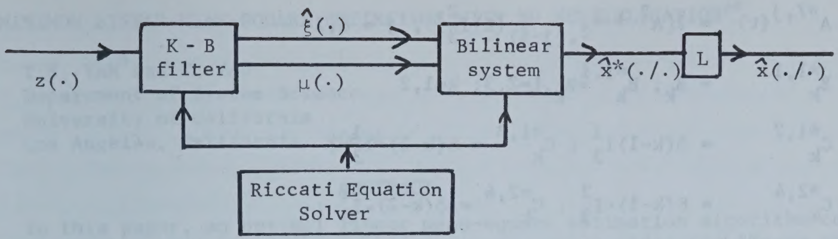


Figure 1

However, a major drawback of the filter is its "curse of dimensionality". The following example illustrates the problem and shows one way of its mitigation. Example: We now apply the algorithm developed in the theorem to construct the optimal filter for the case of system (2.3) with M = 3,

$$N = 2 \text{ and } A = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 & b_1^i & 0 \\ 0 & 0 & b_2^i \\ 0 & 0 & 0 \end{bmatrix}, \quad i = 1,2$$

The filter is found to have a dimension of 21, having required 3 applications of Kushner's equations. We enumerate below the nonzero 3 x 3 blocks of $A^*(t)$, $\{B_k^*\}_{k=1}^{N}$, $\{C_k^*\}_{k=1}^{N}$ and L. The notation used is as follows:

$T^{i,j}$: i,j th block of matrix T

$P_{i,j}(t), \epsilon_{i,j}(t)$: i,j th elements of matrices P(t) and $[F(t)-P(t)H^T(t)R^{-1}(t)H(t)]$ respectively.

δ(·): Krönecker delta function.

and finally, for any n x n matrix T and 0 ≦ i ≦ n,

T^i: Matrix T with last i rows and columns replaced by zeros.

$$A^{*1,1}(t) = A; \; A^{*1,j}(t) = B^1_{j-1}, j=2,3;$$

$$A^{*i,1}(t) = \sum_{k=1}^{2} P_{(i-1),k}(t)B_k; \; A^{*i,j}(t) = [A^1+\epsilon_{(i-1),(j-1)}(t)I_3^1], \quad i,j=2,3;$$

$$A^{*4,2}(t) = 2\sum_{i=1}^{2} B_i^1 P_{1,i}(t), \; A^{*4,j}(t) = 2[A^2+\epsilon_{1,(j-3)}(t)I_3^2], \; j=4,5;$$

$$A^{*i,2}(t) = \sum_{k=1}^{2} P_{2,k}(t)B_k, \; i=5,6; \; A^{*i,3}(t) = \sum_{k=1}^{2} P_{1,k}(t)B_k, \; i=5,6;$$

$$A^{*i,j}(t) = [A^2 + \epsilon_{2,(j-3)}(t)I_3^2], \; i=5,6; \; j=4,5;$$

$$A^{*i,j}(t) = [A^2 + \epsilon_{1,(j-5)}(t)I_3^2], \; i=5,6; \; j=6,7;$$

$$A^{*7,3}(t) = 2\sum_{k=1}^{2} P_{2,k}(t)B_k^1;$$

$$A^{*7,j}(t) = 2[A^2 + \epsilon_{2,(j-5)}(t) I_3^2] , \quad j = 6,7$$

$$B_k^{*1,1} = B_k ; \; B_k^{*i,i} = B_k , i=2,3; \; k=1,2$$

$$C_k^{*1,2} = \delta(k-1) I_3^1 \; ; \; C_k^{*1,3} = \delta(k-2)\cdot I_3^1 \; ;$$

$$C_k^{*2,4} = \delta(k-1)\cdot I_3^2 \; ; \; C_k^{*2,6} = \delta(k-2)\cdot I_3^2 \; ;$$

$$C_k^{*3,5} = \delta(k-1)\cdot I_3^2 \; ; \; C_k^{*3,7} = \delta(k-2) \, I_3^2 \; ; \; k = 1,2$$

$$L^{1,1} = I_3$$

In the above matrices, note the repetition of certain block rows and the presence of some all zero rows. This allows us in effect to reduce the filter dimension from 21 to 10. This argument can be extended to the general case, and it is not difficult to show that the dimensionality can be reduced in this way from

$$\frac{M}{N-1}(N^M-1) \quad \text{to} \quad \sum_{i=0}^{M-1} (M-i) \binom{N+i-1}{i}$$

References:

1. James T. Lo and Alan S. Willsky, "Estimation for Rotational Processes with One Degree of Freedom - Part I: Introduction and Continuous-Time Processes", IEEE Trans. on A.C., 20, 10-21 (1975).
2. James T. Lo, "Signal Detection for Bilinear Systems", Information Sciences, 9, 249-278 (1975).
3. Steven I. Marcus and Alan S. Willsky, "Algebraic Structure and Finite Dimensional Nonlinear Estimation", S.I.A.M. J. on Math. Anal., 9, 312-327 (1978).
4. M. Fliess, "Un Outil Algebrique: Les Series Formelles Non Commutatives", Lecture Notes in Math. Systems and Economics, Springer-Verlag (Proc. of the Conf. on Math. Systems Theory, Udine, Italy, 1975).
5. H. Sussman, "Semigroup Representation, Bilinear Approximation of Input-Output Maps, and Generalized Inputs", Lecture Notes in Math. Systems and Economics, Springer-Verlag. (Proc. Conf. on Math. Systems Theory, Udine, Italy, 1975).
6. Harold J. Kushner, "Dynamical Equations for Optimal Nonlinear Filtering", J. Diff. Equations, 3, 179-190 (1967).
7. R. W. Brockett, "Volterra Series and Geometric Control Theory", Automatica, 12, 167-176 (1976).
8. K. S. Miller, Multidimensional Gaussian Distributions, John Wiley and Sons, New York (1965).
9. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory", Trans. ASME, Ser. D.; J. Basic Eng., 83, 95-107 (1961).
10. T. Kailath and P. Frost, "An Innovations Approach to Least-Squares Estimation: Part II. Linear Smoothing in Additive White Noise", IEEE Trans. on A.C. , 13, 655-660 (1968).

This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-74-2671C.

ON MINIMUM LINEAR MEAN-SQUARE ESTIMATION WITH NO MULTIPLICATION**

T.Y. YAN* and K. YAO
Department of System Science
University of California
Los Angeles, California 90024

ABSTRACT

In this paper, an optimal linear mean-square estimation algorithm constrained to be multiplication-free is derived. We consider the linear estimation problem on a binary computer where the estimation parameters are constrained to be powers of two and thus all multiplications are replaced by single shifts. These integer-programming-like problems are converted by the branch-and-bound principle to a series of convex programming problems. Details on the algorithm for multiplication-free solutions and various numerical examples are given.

I. INTRODUCTION

In recent years, the existence of large scientific computers, as well as special microprocessor hardware, have made digital signal processing attractive for many tasks in communication systems. Since many of these problems require real-time solutions, the selection of appropriate algorithms is particularly important. While there are many factors relevant to selecting the algorithms, in this paper we wish to consider the problem of the elimination of multiplications.

In most small computer systems, the operation of multiplication involves considerably more machine cycles, often one order of magnitude more, than such operations as add, subtract or shift. Yet, in most signal-processing applications, such as estimation, convolution, and filtering, multiplication is certainly a basic required operation. In this paper, we will consider a class of linear mean-square estimation problems with no multiplication.

In using a digital computer for processing analog data, there are the usual A/D quantization errors as well as internal arithmetic round-off errors. On the other hand, the quantization of parameters intrinsic to the processing algorithms can be controlled by the designer. We do not, however, use the "round to the nearest" procedure in forcing the set of real-valued processing parameters to allowable quantized values. While the rounding procedure tends to be good for large word-length registers, this approximation can be non-ideal in many cases. Here, we wish not only to control this parameter quantization effect more precisely, but also to process the data as "quickly" as possible.

---

*T.Y. Yan is presently with Hughes Aircraft Company, Canoga Park, CA.

**This work was supported by the Electronics Program of the Office of Naval Research.

For a linear block estimation problem, we need to perform the multiplication of the processing matrix with the random data vector. From the theory of computational complexity, various results for lower bounds on the number of multiplications of a vector by matrix are known. Such a minimal number of multiplication algorithms may hold little promise of being implemented in practical signal-processing schemes, due to the resulting complexities in the increased real-time and pre-processing programming and storage costs.

A technique that utilizes the structure and properties of discrete programming via the branch-and-bound principle is used to develop algorithms which are constrained to be multiplication-free. These integer-programming-like problems are converted into a series of continuous variable optimization problems. It is shown that, if the objective processing criterion is a definite quadratic functional, the simplex iteration in linear programming can be modified to obtain a unique discrete solution. An explicit dynamically growing binary tree is used as a form of selective information retrieval system to keep track of all possible feasible solutions. Numerical examples are given in demonstrating the usefulness of the discrete multiplication-free algorithm for linear estimation problems.

II. LINEAR MINIMUM MEAN-SQUARE ESTIMATION

Consider the estimation model shown in Fig. 1 below. The signal x and the noise n are two zero-mean real-valued column random vectors of dimensions M and N , respectively. The covariance and cross covariance matrices Λ_{xx} , Λ_{nn} , and Λ_{nx} , as well as the $N \times M$ matrix L, are assumed to be known.

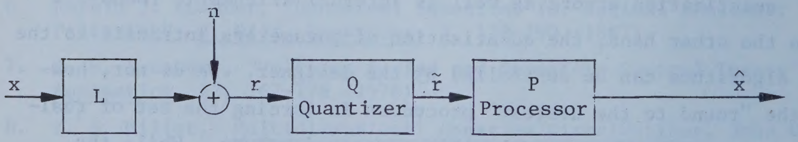


FIG. 1 - Estimation Model

Let us denote the received column data vector by $r = Lx + n$, the quantized data column vector by $\tilde{r}$, the processor by P , and the optimum estimate of x by $\hat{x}$. We assume that full prior statistics are available regarding the probabilistic description of the parameters of this problem. The estimation error can be defined as $\tilde{x} = x - \hat{x}$. Let $C(\tilde{x})$ be the cost function associated with the error. Our goal is to minimize the overall expected cost of the error in estimation by the proper choice of $\tilde{x}$. As usual, let $C(\tilde{x}) = \|\tilde{x}\|^2$. The design of the processor P and the quantizer Q clearly depends on the specific cost function being chosen. In the absence of the quantizer, it is well known that, if x and n are jointly Gaussian distributed, the best estimators under the above cost function is a linear estimator. In general, for ease of implementation as well as ease of analysis, we assume that the processor P is linear with a representation matrix A.

The mean square error

$$\varepsilon \overset{\Delta}{=} E\|x\|^2 = E\{\|x-A\tilde{r}\|^2\}$$

$$= E\{\|[x-Ar] + [A(r-\tilde{r})]\|^2\} \quad . \tag{2.1}$$

We make the standard assumption that the quantization error $q = r-\tilde{r}$ is uncorrelated to the signal and noise. Thus, the two bracketed terms in (2.1) are uncorrelated. Then,

$$\varepsilon = E\,\mathrm{Tr}\{(x-Ar)(x-Ar)' + A(r-\tilde{r})(r-\tilde{r})'A'\}$$

$$= \mathrm{Tr}\{\Lambda_{xx} - 2A\Lambda_{rx} + A(\Lambda_{rr} + \Lambda_{qq})A'\} \quad , \tag{2.2}$$

where $\Lambda_{rx} = L\Lambda_{xx} + \Lambda_{nx}$ and $\Lambda_{rr} = L\Lambda_{xx}L' + \Lambda_{nx}L' + L\Lambda_{xn} + \Lambda_{nn}$ are expressible in terms of previously specified parameters and Λ_{qq} is also assumed to be known from quantizer characterization.

Now, let a_i' be the row vectors of A and b_i be the column vectors of Λ_{rx}, $1 \le i \le M$. Then Eq.(2.2) can be re-expressed as

$$\varepsilon = Q_o + \sum_{j=1}^{M} Q_j(a_j) \quad , \tag{2.3}$$

where $Q_o = \mathrm{Tr}\{\Lambda_{xx}\}$, $Q_j(a_j) = a_j'(\Lambda_{rr} + \Lambda_{qq})a_j - 2a_j'b_j$, $1 \le j \le M$.

We note that Q_o is a constant independent of a_j, while each $Q_j(a_j)$ is a convex functional of a_j in $\mathbb{R}^n$. Furthermore, for physical systems, we can assume $(\Lambda_{rr} + \Lambda_{qq})$ is a positive definite matrix and A and ε will have unique optimum solutions. Specifically, A can be found by setting

$$\nabla_{a_j} \varepsilon = \nabla_{a_j} Q_j(a_j) = 0 \quad , \quad 1 \le j \le M \ .$$

Then the optimum a_j is given uniquely by

$$\hat{a}_j = (\Lambda_{rr} + \Lambda_{qq})^{-1}\, b_j \quad , \quad 1 \le j \le M \quad , \tag{2.4}$$

and the unique global minimum ε is given by

$$\varepsilon_{Min} = Tr\{\Lambda_{xx}\} - \sum_{j=1}^{M} \hat{a}_j' b_j \ . \tag{2.5}$$

The optimum estimate of x is then given by

$$\hat{x} = \hat{A}\tilde{r} = [\hat{a}_1' \tilde{r}, \hat{a}_2' \tilde{r}, \cdots, \hat{a}_M' \tilde{r}]' \ . \tag{2.6}$$

Implementation of $\hat{x}$ in Eq.(2.6) generally requires $M \times N$ multiplications. However, by using the branch-and-bound principle to be discussed in the next section, we can construct the optimum linear processing matrix $\tilde{A}$ subject to the constraint that the magnitude of each element is a power of 2. Thus, the estimation operation of $\tilde{A}\tilde{r}$ corresponding to Eq.(2.6) involves no multiplication.

III. CONCEPT OF BRANCH AND BOUND

Now, we will present a brief description of the branch and bound principle [1] and its application to the linear estimation problem. For this minimization problem, we begin the operation by defining an extended solution space with a modified cost function. The solution space is repeatedly divided into smaller and smaller subsets and a bound is computed for the cost of the solutions within each subset. After each subdivision, those subsets with a bound that exceeds the cost of a known feasible solution are excluded from further operation. This process continues until a feasible solution is found such that its cost is no greater than the bound for any subset. By using this principle, we obtain the following theorem applicable to each of the $Q_j(\cdot)$ quadratic forms in Eq.(2.3) The actual implementation of this theorem is quite involved and the detailed steps are omitted here.

THEOREM The branch and bound algorithm will terminate in a finite number of steps H and the solution x^H is the best discrete multiplication-free solution. That is,

$$Q_j(x^H) = \min_{x \in Z^n} Q_j(x) = \min_{x \in T^H} Q_j(x)$$

where $T^H \subset \mathbb{R}^n$, T^H is the solution space after the H^{th} iteration, Z is the discrete set of allowable magnitude values of x_i constrained to be powers of two, and $x = (x_1, x_2, \cdots, x_n)$.

IV. SPECIFIC EXAMPLES

Now, we will use the branch and bound principle to obtain some specific optimal multiplication-free mean-square estimation solutions.

Example 1. Consider a real-valued linear mean-square estimation problem of Fig. 1, with $L = I_4$, $\Lambda_{xx} = I_4$, $\Lambda_{nn} = I_4$, $\Lambda_{qq} = .001\ I_4$, and $\Lambda_{nx} = \underline{0}$, where I_n is the $n \times n$ identity matrix. By direct evaluation of Eq. (2.4), the optimum a_j is given by

$$\hat{A} = \begin{bmatrix} .4168 & .1709 & .0714 & .0331 \\ .1709 & .3686 & .1540 & .0714 \\ .0714 & .1540 & .3686 & .1709 \\ .0331 & .0714 & .1709 & .4168 \end{bmatrix} \tag{4.1}$$

and the minimum mean-square error $\varepsilon_{Min} = 1.572$. This optimal estimator needs at least six multiplications. If each element of A in eq. (4.1) is "rounded to the nearest" discrete value in the set $D = \{\pm 2^{-1}, \pm 2^{-2}, \pm 2^{-3}, \pm 2^{-4}, \pm 2^{-5}\}$, then the rounded processing matrix is given by

$$A_R = \begin{bmatrix} 2^{-1} & 2^{-3} & 2^{-4} & 2^{-5} \\ 2^{-3} & 2^{-2} & 2^{-3} & 2^{-4} \\ 2^{-4} & 2^{-3} & 2^{-2} & 2^{-3} \\ 2^{-5} & 2^{-4} & 2^{-3} & 2^{-1} \end{bmatrix}$$

and the associated mean-square error $\varepsilon_R = 1.697$. Now, if we use the branch and bound principle, the processing matrix is given by

$$\tilde{A} = \begin{bmatrix} 2^{-1} & 2^{-3} & 2^{-4} & 2^{-5} \\ 2^{-3} & 2^{-1} & 2^{-3} & 2^{-4} \\ 2^{-4} & 2^{-3} & 2^{-1} & 2^{-3} \\ 2^{-5} & 2^{-4} & 2^{-3} & 2^{-1} \end{bmatrix}$$

and the associated mean-square error of $\tilde{A}$ is $\varepsilon_{\tilde{A}} = 1.653$. The value of $\varepsilon_{\tilde{A}}$ is greater than the minimum value ε_{Min}. However, the optimal multiplication-free $\varepsilon_{\tilde{A}}$ is indeed less than the "round to the nearest" ε_R. If we want to reduce $\varepsilon_{\tilde{A}}$ further, we can permit from one to six multiplications in the processing. As expected, as the number of multiplications increases, the associated ε will decrease monotonically to ε_{Min}.

Example 2. Consider the linear mean-square estimation problem of Fig.1, with a first-order Markov signal sequence with covariance matrix

$$\Lambda_{xx} = [\rho^{|i-j|}]_{i,j=1,\cdots,8} \quad ,$$

$L = I_8$, and white covariance $\Lambda_{nn} = \sigma_n^{\ 2} I_8$. Various approaches have been used in the past to reduce the complexity of the linear estimator. All these approaches, however, require a certain amount of overhead programming cost in implementing the associated estimation algorithms. The implementation of our multiplication-free linear estimator is extremely simple since only single shifts are needed. In all our results, Λ_{qq} is evaluated by using that of a 6-bit Max quantizer which is optimal in the sense of introducing a minimum mean-square quantization error for Gaussian inputs. Fig.2 shows the mean-square error of the multiplication-free solution with block length $N \doteq 8$ and correlation coefficient $\rho = .7$, for various processing magnitude bits B. For each SNR, the branch and bound principle yields an optimal finite number of processing magnitude bits B_{opt}. As seen in Fig.2, if the actual B used is much smaller than B_{opt}, there can be a significant increase in the mean-square error. For B closed to B_{opt}, however, the difference between the multiplication-free solution and the classical solution is quite small. Fig.3 shows the mean-square error for data of block lengths $N = 8$ at $\mathrm{SNR} = 10 \log (1/\sigma_n^2)$ of 0 dB for different ρ. The solid curves represent the classical minimum mean-square solutions while the dashed curves represent the optimal 6-bit quantizer multiplication-free solutions.

V. CONCLUSION

We have used the branch and bound principle to obtain an optimal multiplication-free solution for the linear mean-square estiamtion problem. Our numerical results tend to indicate that the mean-square error associated with the multiplication-free solution is not significantly greater than that of the classical solution which uses full precision multiplication. Thus, the replacement of multiplications by single shifts, at least for these estimation problems, appears to be a simple and useful approach to attack the practical problems of computational complexity for real-time operating systems. The technique described here can be used to obtain useful multiplication-free solutions for several other communication and information-processing problems.

REFERENCE

1. L.G. Mitten, "Branch-and-Bound Methods: General Formulation and Properties," *Opns.Res.*, *18* (1970) 24-34.

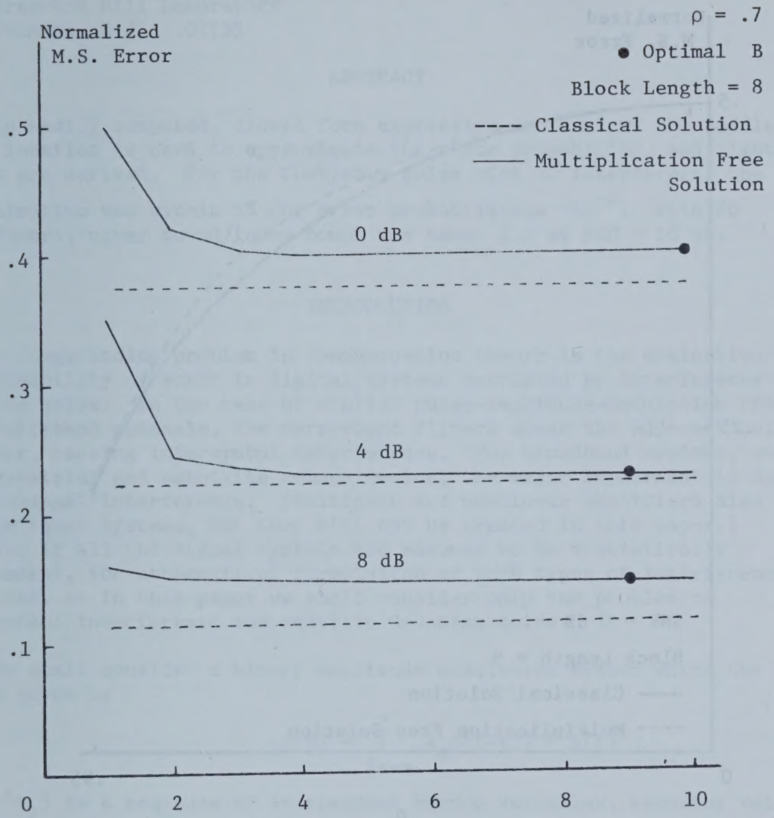


FIG. 2 Normalized Mean-Square Error of the First Order Markov Sequence in White Noise

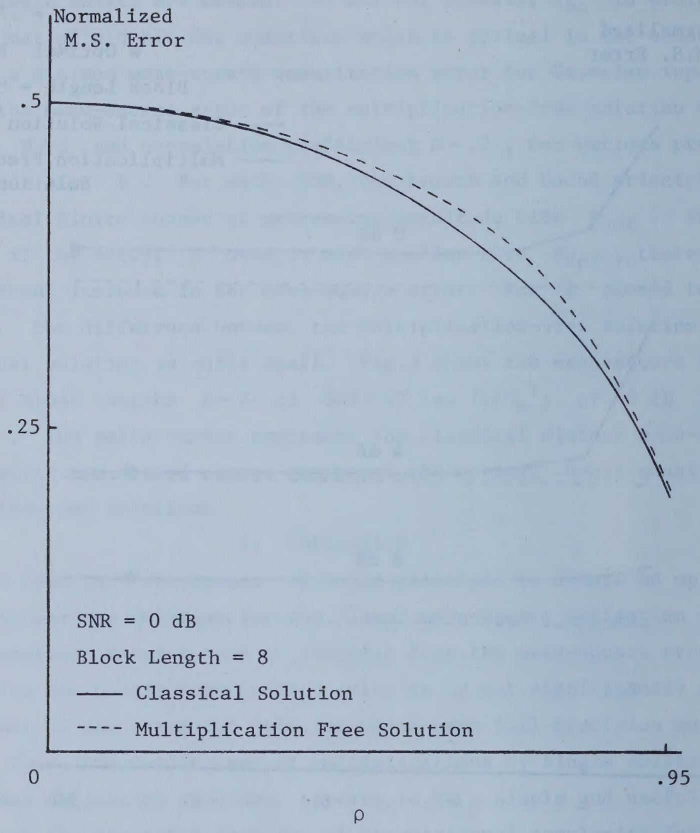


FIG. 3 Normalized Mean-Square Error for the Correlation Coefficient of the First Order Markov Sequence in White Noise

SADDLE POINT APPROXIMATION FOR THE ERROR PROBABILITY IN PAM SYSTEMS WITH INTERSYMBOL INTERFERENCE

ON-CHING YUE
Bell Laboratories
Crawford Hill Laboratory
Holmdel, N.J. 07733

ABSTRACT

An easily computed, closed form expression in terms of the saddle point location is used to approximate the error probability, and tight bounds are derived. For the Chebyshev pulse with 10 interferers, the approximation was within 1% for error probabilities $<10^{-6}$. With 80 interferers, upper bound/lower bound was about 1.1 at SNR = 16 dB.

INTRODUCTION

A longstanding problem in communication theory is the evaluation of the probability of error in digital systems corrupted by interference and additive noise. In the case of digital pulse-amplitude-modulation (PAM) over voiceband channels, the narrowband filters smear the adjacent pulses together, causing intersymbol interference. For broadband systems, such as terrestrial and satellite communication, the major impairment is due to co-channel interference. (Multipath and nonlinear amplifiers also degrade these systems, but they will not be treated in this paper.) However, if all the signal symbols are assumed to be statistically independent, the mathematical formulation of both types of interference are identical, so in this paper we shall consider only the problem of intersymbol interference and additive Gaussian noise.

We shall consider a binary amplitude modulation system which has an output given by

$$y(t) = \sum_{\ell=-\infty}^{\infty} a_\ell r(t-\ell T) + n(t) \ , \tag{1}$$

where $\{a_\ell\}$ is a sequence of independent random variables, assuming values +1 and -1 with equal probability. $r(t)$ is the impulse response of the overall system and $n(t)$ is assumed to be zero-mean with variance σ^2, Gaussian, and independent of the a_ℓ's.

The detector samples $y(t)$ every T seconds to recover the transmitted sequence. The sampled signal at time t_o is

$$y(t_o) = a_o r(t_o) + \sum_{\ell=-L}^{L}{}' a_\ell r(t_o-\ell T) + n(t_o) \ , \tag{2}$$

where the prime indicates the omission of the zeroth term in the summation. In (2), we have assumed that there are 2L interferers, located on both sides of $a_o r(t_o)$, the desired signal. To simplify the notation, we shall rewrite (2) as

$$\begin{aligned} y_o &= a_o r_o + \Sigma' \, a_\ell r_\ell + n_o \\ &= a_o r_o + R + n_o \, , \end{aligned} \tag{3}$$

where the limits of the summation are understood to be $\ell = -L$ to L. Since the a_ℓ's and n_o have probability densities symmetrical about zero, it can be shown that the optimum threshold level in the detector for minimizing the error probability is zero. Then the minimum error probability is given by

$$P_e = \Pr\{R+n_o > |r_o|\} \, . \tag{4}$$

Since the computational time of the direct, exhaustive method of calculating the error probability grows exponentially with the number of interferers, various methods have been proposed to estimate the error probability. The results of these different approaches can be divided into two categories as in [1]: infinite series and computer algorithms for approximating the error probability to any degree of accuracy, and simple upper and lower bounds. In the first category there are the Hermite series [2], the Gram-Charlier series [3], and other approaches including the Gauss-Quadrature numerical technique [4]. Although all the techniques in this category give very accurate results, the computational effort required has limited their usefulness.

In the second category there are the worst case [5] and the Chernoff [6] upper bounds which are not tight. Tighter upper and lower bounds were obtained as functions of the interference variance [7] and the fourth moment [8]. It was observed [1] that these bounds became less effective for large signal-to-noise ratios ($\mathrm{SNR} = -10 \log_{10}\sigma^2$). The moment space bounds [9] are very tight for larger SNR, but a parameter had to be optimized via a complicated transcendental equation. The latest result in this area is a lower bound [10] which is a weighted sum of the tail distributions whose arguments are partial sums of the ranked interferers. Twice the lower bound was given as an approximation, but no upper bound was given.

THE SADDLE POINT APPROXIMATION

In order to apply the saddle point method [11] to calculate the error probability, we rewrite (4) as an inverse Fourier integral

$$P_e = \frac{1}{2\pi} \int_C \frac{1}{iu} \exp[-iur_o] \Phi(u) du \, , \tag{5}$$

where the path of integration C can be any contour whose real part goes from $-\infty$ to ∞ and whose imaginary parts lies in the lower half of the complex u plane and $\Phi(u)$ is the characteristic function of $R+n_o$ given by

$$\Phi(u) = \exp[-\sigma^2 u^2/2] \prod_{\ell=-L}^{L}{}' \cos(r_\ell u) \, . \tag{6}$$

We define

$$G(u) = -iur_o + \ell n\Phi(u), \tag{7}$$

then (5) can be rewritten as

$$P_e = \frac{1}{2\pi}\int_C \frac{1}{iu}\exp[G(u)]du . \tag{8}$$

It can be shown that the integrand in (8) is analytic in the entire complex u-plane except for the origin, so that we can choose the contour of integration to be any horizontal line parallel to, but below the real axis; the contribution by the two segements at ±∞ connecting to the real axis is zero by the Riemann-Lebesgue Theorem. The saddle points of the integrand (ignoring the effect of u^{-1}) are solutions of $G'(u) = 0$, i.e.,

$$r_o = iu\sigma^2 + i\Sigma' r_\ell \tan r_\ell u . \tag{9}$$

It can be shown that there is always a unique solution of (9) on the imaginary axis. Denoting it by -ib, we can rewrite the saddle point equation as

$$r_o = b\sigma^2 + \Sigma' r_\ell \tanh r_\ell b = S(b) . \tag{10}$$

Let the contour of integration for (8) intersect the imaginary axis at u = -ib. Then expanding G(u) around u = -ib, we have

$$G(u) = G_b - \frac{(u+ib)^2}{2}\sigma_b^2 + \frac{(u+ib)^3}{6}G'''(-ib) + \ldots,$$

where

$$G_b = -br_o + \frac{b^2\sigma^2}{2} + \Sigma' \ln\cosh r_\ell b , \tag{11}$$

and

$$\sigma_b^2 = \sigma^2 + \Sigma' r_\ell^2 \operatorname{sech}^2 r_\ell b . \tag{12}$$

We define a new variable of integration x = u + ib, then (8) can be rewritten as

$$P_e = \frac{\exp[G_b]}{2\pi}\int_{-\infty}^{\infty} \exp[-\frac{x^2}{2}\sigma_b^2]\cdot[1 + \frac{x^3}{6}G'''(-ib)+\ldots]\frac{dx}{b+ix} .$$

Integrating term by term, we obtain formally an infinite series representation of P_e, with the leading term given by

$$\hat{P}_e = \frac{1}{2} \exp[G_b + \frac{b^2}{2} \sigma_b^2] \operatorname{erfc}[b\sigma_b/\sqrt{2}] \ . \tag{13}$$

We see that $\hat{P}_e$, the saddle point approximation, is a closed form expression in terms of b, the saddle point location. It can be shown that the approximation is asymptotically correct for large b, which corresponds to small σ^2 and small r_ℓ's.

Differentiating the saddle point equation (10) twice to obtain

$$S''(y) = -\Sigma' 2r_\ell^3 \tanh r_\ell y \operatorname{sech}^2 r_\ell y \ ,$$

we see that $S''(y)$ does not change sign in $0 \leq y < \infty$, which means Newton's root-finding algorithm for solving for b:

$$b_{k+1} = b_k + [r_o - S(b_k)]/S'(b_k)$$

would converge monotonically, and a good initial guess is $b_o = 0$. In the example of the Chebyshev pulse, it took three to four iterations to determine b to 5-place accuracy.

UPPER AND LOWER BOUNDS

To obtain bounds for P_e, we define a new integral:

$$\begin{aligned} J &\triangleq P_e \cdot b \exp[-G_b] \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp[G(x-ib)-G_b] \frac{dx}{1+i\frac{x}{b}} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp[-H_R(x)+iH_I(x)] \frac{dx}{1+i\frac{x}{b}} \ , \end{aligned} \tag{14}$$

where

$$H_R(x) = \frac{x^2\sigma^2}{2} - \frac{1}{2} \Sigma' \ln [\cos^2 r_\ell x + \sin^2 r_\ell x \cdot \tanh^2 r_\ell b],$$

and

$$H_I(x) = -xr_o + b\sigma^2 x + \Sigma' \arctan [\tan r_\ell x \cdot \tanh r_\ell b] \ .$$

We shall derive upper and lower bounds for J by bounding $H_R(x)$ and $H_I(x)$.

It can be shown that

$$H_R(x) \geq \frac{x^2\sigma^2}{2} \ , \quad \text{for all } x \ ,$$

and

$$H_R(x) \geq \frac{x^2}{2}\sigma_u^2, \quad \text{for } 0 \leq x \leq X_u ,$$

where

$$\sigma_u^2 = \sigma^2 + \frac{1}{2}\Sigma' r_\ell^2 \operatorname{sech}^2 r_\ell b ,$$

and

$$X_u^2 = \frac{3\Sigma' r_\ell^2 \operatorname{sech}^2 r_\ell b}{2\Sigma' r_\ell^4 \operatorname{sech}^2 r_\ell b} .$$

Then an upper bound for J is given by

$$J_u = (2\pi\sigma_u^2)^{-1/2} \operatorname{erfc}[X_u \sigma_u/\sqrt{2}]$$

$$+ (2\pi\sigma^2)^{-1/2} \operatorname{erfc}[X_u \sigma/\sqrt{2}] \cdot \left[1 + \left(\frac{X_u}{b}\right)^2\right]^{-1/2} . \qquad (15)$$

For $0 \leq x \leq X_R$, we can show that

$$H_R(x) \leq \frac{x^2}{2}\sigma_\ell^2 , \qquad (16)$$

where

$$X_R = 0.75 \operatorname*{Min}_{\ell \neq 0} \left\{\frac{1}{r_\ell \operatorname{sech} r_\ell b}\right\} ,$$

and

$$\sigma_\ell^2 = \sigma^2 + \frac{3}{2}\Sigma' r_\ell^2 \operatorname{sech}^2 r_\ell b .$$

By noting that $H_I(x)$ is convex U for $0 \leq x \leq X_I = \pi/2 \operatorname*{Max}_{\ell \neq 0}\{r_\ell\}$, we have

$$H_I(x) \leq \frac{H_I(X_I)}{X_I} x \triangleq Bx . \qquad (17)$$

Let $X_\ell = \min\{X_r, X_i, \sqrt{2}/B\}$, then we can define a truncated version of (14):

$$J_T = \frac{1}{2\pi}\int_{-X_\ell}^{X_\ell} \exp[-H_R(x) + iH_I(x)] \frac{dx}{1 + i\frac{x}{b}} ,$$

which can be bounded from below by using (16) and (17). Bounding the difference $|J-J_T|$ from above, we have a lower bound for J:

$$J_\ell = (2\pi\sigma_\ell^2)^{-1/2}\,\mathrm{erf}\{X_\ell\sigma_\ell/\sqrt{2}\}\cdot\left[1-\left(\frac{B^2}{2}+\frac{1}{b^2}\right)\frac{1}{\sigma_\ell^2}+\frac{3B^2}{2b^2\sigma_\ell^4}\right]$$

$$+\frac{X_\ell}{\pi\sigma_\ell^2}\exp[-X_\ell^2\sigma_\ell^2/2]\cdot\left[\frac{B^2}{2}+\frac{1}{b^2}-\frac{3B^2}{2b^2\sigma_\ell^2}\left(1+\frac{X_\ell^2\sigma_\ell^2}{3}\right)\right]$$

$$-(2\pi\sigma^2)^{-1/2}\,\mathrm{erfc}[X_\ell\sigma/\sqrt{2}]\cdot\left[1+\left(\frac{X_\ell}{b}\right)^2\right]^{-1/2}. \qquad (18)$$

AN EXAMPLE

To illustrate the accuracy of the saddle point approximation we consider the example of the fourth-order Chebyshev pulse as defined by:

$$r(t) = A_1\cos(\omega_1|t| - \theta_1)\cdot\exp[-\alpha_1|t|]$$

$$+ A_2\cos(\omega_2|t| - \theta_2)\cdot\exp[-\alpha_2|t|]$$

with

$$A_1 = 0.4023, \qquad A_2 = 0.7162,$$
$$\omega_1 = 2.839, \qquad \omega_2 = 1.167,$$
$$\theta_1 = 0.7553, \qquad \theta_2 = 0.1602,$$
$$\alpha_1 = 0.4587, \qquad \alpha_2 = 1.107.$$

First we let L = 5, so that exact values of the error probability can be computed. As in [6], the noise variance is chosen to be $\sigma^2 = (6.361)^{-2}$ giving an error probability of 10^{-10} in the absence of intersymbol interference. We also let T = 1. For different values of the sampling time t_o, measured as deviation from the nominal value, we found that the saddle point approximation ($\hat{P}_e$) differ from the exact one (P_e) by less than 5 percent, so in Figure 1 we plot $\hat{\varepsilon} = \ln(\hat{P}_e/P_e)$. Note that for small $\hat{\varepsilon}$, $\hat{\varepsilon} \cong (\hat{P}_e-P_e)/P_e$, the fractional error. As t_o decreases from 0.2 to 0, the saddle point location b increases from about 13 to 36 and, as shown in the figure, $\hat{\varepsilon}$ approaches zero as expected from the asymptotic nature of $\hat{P}_e$. At

$t_o = 0.2$, which corresponds to $P_e \cong 10^{-3}$, $|\hat{\varepsilon}| < 0.04$ and at $t_o = 0.1$, corresponding to $P_e \cong 10^{-6}$, $|\hat{\varepsilon}| < 0.01$. For comparison the approximation given by Jenq [10] was also included in Figure 1.

In Figure 2 we show the upper and lower bounds normalized by P_e vs. t_o. For $t_o < 0.1$, the two bounds are within 15 percent of each other. As t_o increases, they are farther apart, and at $t_o = 0.2$ the upper bound is about 2-1/2 times the lower bound. Jenq's lower bound lies between 40 to 60 percent of the exact value.

The various moment space bounds were tabulated in Table II of reference [9] for the case of L = 40, as a function of signal-to-noise ratio, SNR = 20 log (r_o/σ), with $r(t_o=0) = 1$. We have taken the variance and the optimum exponential moment bounds and compared them with the saddle point approximation and the associated bounds in Table I. As noted in [1], the variance bounds are tight for small SNR, corresponding to error probabilities $> 10^{-4}$, but become poor as SNR increases. The exponential moment bounds are uniformly tight for the range of σ^2 shown. While the saddle point bounds are as good as the exponential moment ones for moderate to large SNR, they diverges to a ratio of 10 to 1 for SNR = 0 dB. Even though the saddle point approximation is designed to perform well for large SNR, it is surprisingly accurate even for small SNR.

CONCLUSIONS

In this paper we have derived the saddle-point approximation for the error probability in binary PAM systems with intersymbol interference and additive Gaussian noise. The result is a closed form expression, in terms of the saddle point location which can be accurately determined after several iterations. Tight upper and lower bounds were also obtained. In the example of the Chebyshev pulse, with 10 interferers, the approximation was within several percents of the exact value. With 80 interferers the bounds were as good as the exponential moment bounds for moderate to large SNR, but involving less computation.

The saddle point method can be applied to more complicated interference problems, such as correlated signal symbols and intersymbol plus co-channel interference, as long as the characteristic function of the interferers is known in closed form.

REFERENCES

1. P. J. MacLane, "Upper and Lower Bounds for Error Probability in Pulse Communication Systems", Proc. 13th Annual Allerton Conf. on Circuit and System Theory, pp. 782-791, October, 1974.

2. V. K. Prabhu, "Error Rate Considerations for Coherent Phase-Shift Keyed Systems with Co-channel Interference", Bell Syst. Tech. Jour., Vol. 48, pp. 743-767, March, 1969.

3. O. Shimbo and M. J. Celebiler, "The Probability of Error Due to Intersymbol Interference and Gaussian Noise in Digital Communication Systems", IEEE Trans. Commun., Vol. COM-19, pp. 113-120, April, 1971.

4. S. Benedetto, G. de Vincentiis and A. Luneson, "Error Probability in the Presence of Intersymbol Interference and Additive Noise for Multilevel Digital Signals", IEEE Trans. on Comm., vol. COM-21, pp. 181-190, March, 1973.

5. R. W. Lucky, J. Salz and E. J. Weldon, Jr., Principle of Data Communication, New York, McGraw-Hill, 1968, p. 65.

6. R. Lugannani, "Intersymbol Interference and Probability of Error in Digital System", IEEE Trans. Inform. Theory, vol. IT-15, pp. 682-688, November, 1969.

7. J. W. Matthews, "Sharp Error Bounds for Intersymbol Interference", IEEE Trans. Inform. Theory, vol. IT-19, pp. 440-447, July, 1973.

8. P. J. McLane, "Lower Bounds for Finite Intersymbol Interference Error Rates", IEEE Trans. on Commun., vol. COM-22, pp. 853-857, June, 1974.

9. K. Yao and R. M. Tobin, "Moment Space Upper and Lower Bounds for Digital Systems with Intersymbol Interference", IEEE Trans. Inform. Theory, vol. IT-22, pp. 65-74, January, 1976.

10. Y. C. Jenq, B. Liu, and J. B. Thomas, "Probability of Error in PAM Systems with Intersymbol Interference and Additive Noise", IEEE Trans. Inform. Theory, vol. IT-23, pp. 575-582, September, 1977.

11. G. F. Carrier, M. Krook and C. E. Pearson, Functions of a Complex Variable, New York, McGraw-Hill, 1966, Chapter 6.

TABLE I

Error Probability Bounds for Chebyshev Pulse with 80 Interferers

SNR(dB)	Lower Bounds			Saddle Point Approx.	Upper Bounds		
	Variance	Exponential Moment	Saddle Point		Saddle Point	Exponential Moment	Variance
0	1.593×10^{-1}	1.593×10^{-1}	2.447×10^{-2}	1.593×10^{-1}	2.437×10^{-1}	$1.594\ 10^{-1}$	1.593×10^{-1}
4	5.772×10^{-2}	5.761×10^{-2}	4.393×10^{-2}	5.774×10^{-2}	7.368×10^{-2}	5.783×10^{-2}	5.773×10^{-2}
8	6.75×10^{-3}	6.69×10^{-3}	6.29×10^{-3}	6.76×10^{-3}	7.71×10^{-3}	6.82×10^{-3}	6.84×10^{-3}
12	6.14×10^{-5}	6.29×10^{-5}	6.23×10^{-5}	6.62×10^{-5}	7.16×10^{-5}	6.84×10^{-5}	1.05×10^{-4}
16	1.28×10^{-9}	3.58×10^{-9}	3.94×10^{-9}	4.19×10^{-9}	4.44×10^{-9}	4.44×10^{-9}	1.04×10^{-7}
20	5.01×10^{-21}	1.25×10^{-17}	1.46×10^{-17}	1.50×10^{-17}	1.56×10^{-17}	1.56×10^{-17}	1.30×10^{-14}

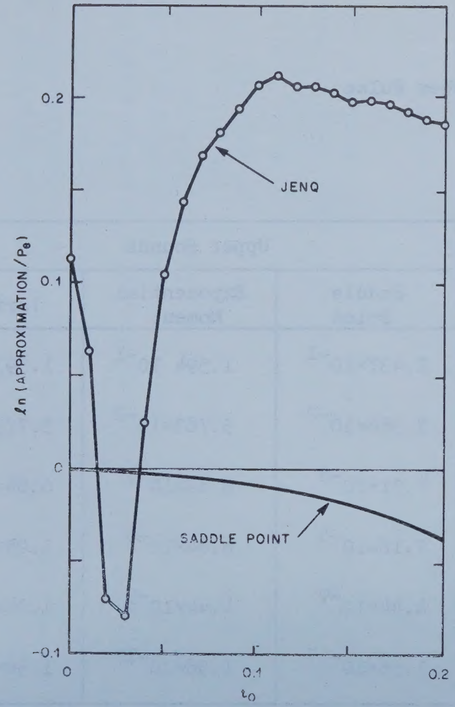


Fig. 1 - Comparison of the Saddle Point Approximation and the Jenq [10] result for the Chebyshev Pulse with 10 Interferers.

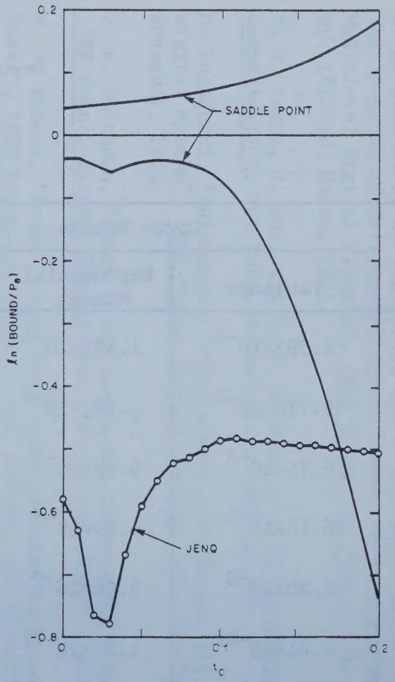


Fig. 2 - Comparison of the Saddle Point Upper and Lower Bounds, and the Jenq [10] lower bound for the Chebyshev Pulse with 10 Interferers.

A NOVEL APPROACH FOR DESIGNING NONLINEAR DISCRETE TIME FILTERS: PART I

D. MINOO-HAMEDANI and G.L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

and

N.C. GALLAGHER and T.E. McCANNON
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 49707

ABSTRACT

The problem of minimum mean squared error prediction of a discrete time random process using a nonlinear filter consisting of a zero memory nonlinearity followed by a linear filter is studied. Classes of random processes for which the best predictor is realizable using a nonlinear filter of the above form are discussed. For those random processes for which the best predictor is not realizable using the above nonlinear filter, an iterative procedure is presented for finding a suboptimal nonlinear filter.

I. INTRODUCTION

In this paper we consider a second order random process $\{X_n, n=1,2,...\}$, and we are interested in predicting the random variable X_{N+1} from an observation of $X_1,...,X_N$. Our estimate is denoted by $\hat{X}_{N+1}$, and we wish to choose it so as to minimize the mean squared error.

It is well known [1, pp.77-78] that the optimal estimate of X_{N+1} in terms of $X_1,...,X_N$ is given by the conditional expectation

$$\hat{X}_{N+1} = E\{X_{N+1} \mid X_N,...,X_1\} .$$

In general, this is a Borel measurable function of $X_1,...,X_N$, and in many cases an exact expression for this quantity is difficult to obtain. Often we do not have the necessary statistical information to evaluate such a quantity. Linear estimation has been widely studied [2], and it is well known that the best linear estimate of X_{N+1} given the observations $X_1,...,X_N$ is obtained by applying the Projection Theorem [1, pp.150-155]. It is clear that in this case the only statistical information required is the second moment characteristics of the random process.

In this paper we restrict our estimate $\hat{X}_{N+1}$ to be of a form that is expressible as the output of a system consisting of a zero memory nonlinearity (ZNL) followed by a linear filter. The ZNL is characterized by a Borel measurable function $g(\cdot)$ such that $g(X_1),...,g(X_{N+1})$ are second order random variables. If the weighting sequence of the linear filter is given by $h_0,...,h_{N-1}$, then the estimate is given by

$$\hat{X}_{N+1} = \sum_{n=1}^{N} g(X_n)h_{N-n} . \qquad (1)$$

We wish to determine a function $g(\cdot)$ and a set of coefficients $h_0,\ldots,h_{N-1}$ in such a way that the resulting mean squared error is minimized. With this form of an estimate, we are guaranteed that the performance can be at least as good as that of the optimal linear filter.

In Section II we consider some cases where the optimal estimate has the form of Eq.(1). In the general case the optimal predictor will not have the form of Eq.(1) and thus a predictor of this form will be suboptimal. This situation is discussed in Section III where an iterative scheme is presented for determining suboptimal predictors. In Section IV examples are given to illustrate the method.

II. OPTIMAL PREDICTION

In this section we consider some cases where the optimal filter has the form of Eq.(1). Whenever the optimal filter is linear, then it obviously has the form of Eq.(1) with $g(x)=x$. The class of spherically invariant random processes [3] admits linear solutions, with the most well-known examples being the Gaussian processes.

It is clear that the performance of the filter given by Eq.(1) can always be made at least as good as that of the optimal linear filter. In some cases the filter given by Eq.(1) can be optimal while the optimal linear filter is useless. For example, let $X_n=P_n(U)$ where U is a random variable uniformly distributed over $[-1,1]$ and $P_n(\cdot)$ is the n-th Legendre polynomial. In this case, the sequence $\{X_n, n=1,2,\ldots\}$ is a sequence of uncorrelated zero mean random variables and the optimal linear filter yields an estimate which is zero. However, for $g(x)=P_{N+1}(x)$ and

$$h_n = \begin{cases} 1, & n=0 \\ 0, & n\neq 0 \end{cases},$$

the filter of Eq.(1) gives the estimate $\hat{X}_{N+1}=X_{N+1}$. Numerous examples similar to this can easily be constructed.

When the process is a (first order) Markov process it is well known [1, pp.81-83] that $E\{X_{N+1}|X_N,\ldots,X_1\} = E\{X_{N+1}|X_N\}$, with probability one (wpl). Thus a system of the form of Eq.(1) with a ZNL given by $g(x)=E\{X_{N+1}|X_N = x\}$ and a weighting sequence given by

$$h_n = \begin{cases} 1, & n=0 \\ 0, & n\neq 0 \end{cases}$$

will yield the optimal estimate of X_{N+1}.

Markov processes serve as the model of many physical phenomena that arise in practice. Often they are obtained as the solution of first order stochastic difference equations of the form

$$X_{n+1} = g(X_n) + Z_{n+1} \quad , \quad n=0,1,2,\ldots$$

where $g(\cdot)$ is a Borel measurable function and the sequence $\{Z_n\}$ is a sequence of zero mean independent random variables independent of the initial condition X_0. It is easily seen that in this case we will have $E\{X_{N+1}|X_N,\ldots,X_1\}=g(X_N)$ wpl.

It is clear that for any random process for which

$$E\{X_{N+1}|X_N,\dots,X_1\} = \sum_{n=1}^{N} g(X_n)h_{N-n} \quad \text{wp1}, \tag{2}$$

a system of the form of Eq.(1) will produce the optimal estimate of X_{N+1}. As another example of a process for which the conditional expectation has the form of Eq.(2) consider the process generated by the following second order stochastic difference equation:

$$X_{n+2} = h_0 g(X_{n+1}) + h_1 g(X_n) + Z_{n+2} , \quad n=-1,0,1,2,\dots, \tag{3}$$

where $g(\cdot)$ is a Borel measurable function and $\{Z_n\}$ is a sequence of zero mean independent random variables independent of the initial conditions X_{-1} and X_0. It can be easily seen that for this example, for any $N \geq 2$,

$$E\{X_{N+1}|X_N,\dots,X_1\} = h_0 g(X_N) + h_1 g(X_{N-1}) \quad \text{wp1}.$$

Extension of this example to the case where Eq.(3) is a k-th order stochastic difference equation is obvious.

To obtain a characterization of a random process for which a form of Eq.(2) holds, we use a theorem due to Balakrishnan [4].

Theorem (Balakrishnan): Let $C_{N+1}(t_1,\dots,t_{N+1})$ denote the joint characteristic function of the random variables $X_1,\dots,X_{N+1}$. Assume that the moments of all orders of the random variables exist, so that $C_{N+1}(\dots)$ has derivatives of all orders. Let D_k denote the differential operator $\partial(\cdot)/\partial it_k$, so that

$$D_k C_{N+1}(t_1,\dots,t_{N+1}) = \frac{\partial}{\partial it_k} C_{N+1}(t_1,\dots,t_{N+1}) .$$

Let $P(x_1,\dots,x_N)$ be a polynomial in N variables. Then a necessary and sufficient condition for

$$E\{(X_{N+1})^M|X_N,\dots,X_1\} = P(X_1,\dots,X_N) \quad \text{wp1}$$

is that

$$\frac{\partial^M}{\partial(it_{N+1})^M} C_{N+1}(t_1,\dots,t_{N+1})\Big|_{t_{N+1}=0} = P(D_1,\dots,D_N)\cdot C_{N+1}(t_1,\dots,t_N,0).$$

Now, in the above theorem let M=1 and let $g(\cdot)$ be a polynomial of degree d, i.e.

$$g(x) = \sum_{j=0}^{d} a_j x^j , \tag{4}$$

and assume $P(x_1,\dots,x_N)$ has the form

$$P(x_1,\ldots,x_N) = \sum_{n=1}^{N} h_{N-n} g(x_n) = \sum_{n=1}^{N} \sum_{j=0}^{d} h_{N-n} a_j (x_n)^j \quad .$$

Assume that the random variables in the process possess moments of all orders. Then a necessary and sufficient condition for Eq.(2) to hold, where $g(\cdot)$ is given by Eq.(4), is that

$$\frac{\partial}{\partial(it_{N+1})} C_{N+1}(t_1,\ldots,t_{N+1})\Big|_{t_{N+1}=0} = \sum_{n=1}^{N} \sum_{j=0}^{d} h_{N-n} a_j D_n^j C_{N+1}(t_1,\ldots,t_N,0) \quad .$$

This result is of limited practical usefulness, because one often does not have the necessary statistical information available.

III. SUBOPTIMAL PREDICTION

In the general case there will not exist a function $g(\cdot)$ and a weighting sequence $h_0,\ldots,h_{N-1}$ such that Eq.(2) is satisfied. However, it is quite reasonable to conjecture that in many cases it may be possible to determine a filter having the form of Eq.(1) with a mean squared error either significantly smaller than that associated with the optimal linear filter or very close to the mean squared error associated with the optimal filter.

Once we assume that the function $g(\cdot)$ that minimizes the mean squared error is known, the $g(X_n)$'s will be well defined random variables and the determination of the h_n's that minimize the mean squared error reduces to an application of the Projection Theorem, i.e. setting

$$E\left\{\left[X_{N+1} - \sum_{n=1}^{N} h_{N-n} g(X_n)\right] g(X_j)\right\} = 0 \ , \quad j=1,\ldots,N,$$

and solving for the h_n's. To carry out this step we need to calculate the terms $E\{g(X_n)g(X_j)\}$ and $E\{X_{N+1}g(X_j)\}$. The difficult problem is the determination of the function $g(\cdot)$ that minimizes the mean squared error.

Notice that, in the optimization problem where the filter is constrained to be of the form in Eq.(1), only second order information (i.e. the family of bivariate distributions) is required. This is more statistical information than is required if we were doing optimal linear filtering, which requires second moment information. However, it is still considerably less statistical information than is required if we were doing optimal filtering, which requires statistical information pertaining to an (N+1)-st dimensional distribution.

In order to circumvent the difficult problem of determining the function $g(\cdot)$ to use in Eq.(1), we will parameterize $g(\cdot)$ and thus let the determination of $g(\cdot)$ simply depend upon finding the correct parameters. Doing so, we would then write the resulting mean squared error as a function of the parameters associated with $g(\cdot)$ and the weighting sequence of the linear filter. In this case, the mean squared error would be a function of K+N parameters, where K is the number of parameters associated with $g(\cdot)$. For example, let $g(\cdot)$ be given by

$$g(x) = \sum_{j=1}^{K} a_j b_j(x) \quad .$$

Then our estimate is given by

$$\hat{X}_{N+1} = \sum_{n=1}^{N} \sum_{j=1}^{K} h_{N-n} a_j b_j (X_n) ,$$

and the resulting mean squared error is given by

$$E\left\{\left[X_{N+1}-\hat{X}_{N+1}\right]^2\right\} = E\left\{\left[X_{N+1}\right]^2\right\} - 2 \sum_{n=1}^{N} \sum_{j=1}^{K} h_{N-n} a_j E\left\{X_{N+1} b_j (X_n)\right\} \qquad (5)$$

$$+ \sum_{n=1}^{N} \sum_{m=1}^{N} \sum_{j=1}^{K} \sum_{k=1}^{K} h_{N-n} h_{N-m} a_j a_k E\left\{b_j (X_n) b_k (X_m)\right\} .$$

The functions $b_j(\cdot)$ should be determined so that there is considerable flexibility in the functional form of $g(\cdot)$ and also so that the expectations in Eq.(5) could be determined from the statistical information at hand. For example, if $b_j(x)=x^j$, then the necessary statistical information would consist of the higher order joint moments.

The next step might be to minimize Eq.(5) over the N+K parameters. This would result in N+K equations of third order polynomials in the parameters. This simultaneous optimization over all the parameters presents potential numerical problems. As an alternative to the simultaneous optimization over all the parameters, we will now describe an iterative technique.

The basic plan of the iterative technique is to consider the two sets of parameters separately and to iteratively optimize over one set of parameters while holding the other set fixed. This iterative technique results in the need to solve systems of linear equations, as opposed to the need to solve systems of equations in third order polynomials such as encountered in the effort to simultaneously optimize over all the parameters.

We will assume that the parametric form of $g(\cdot)$ is such that with the proper choice of parameters we could have $g(x)=x$. In this way the mean squared error that results will always be upper bounded by the mean squared error associated with the optimal linear filter.

The iterative technique is as follows:

Step 1. Determine the optimal weighting sequence $h_0,\ldots,h_{N-1}$ for the case where $g(x)=x$.

Step 2. Evaluate the resulting mean squared error.

Step 3. For this choice of $h_0,\ldots,h_{N-1}$, determine $a_1,\ldots,a_K$ so as to minimize the mean squared error.

Step 4. For this choice of $a_1,\ldots,a_K$, determine the optimal weighting sequence $h_0,\ldots,h_{N-1}$.

Step 5. Repeat Steps 3 and 4 until the improvement in the mean squared error is negligible.

The $a_1,\ldots,a_K$ and $h_0,\ldots,h_{N-1}$ that are obtained in Step 5 after the termination of the iterations determine the system. Step 1 and Step 4 make use of the Projection Theorem and result in $E\{X_{N+1}g(X_j)\} =$

$\sum_{n=1}^{N} h_{N-n} E\{g(X_n)g(X_j)\}$, $j=1,\ldots,N$. Step 2 makes use of Eq.(5). Step 3 also makes use of Eq.(5) and results in

$$\sum_{n=1}^{N}\sum_{m=1}^{N} h_{N-n}h_{N-m}\left[2a_j E\{b_j(X_n)b_j(X_m)\} + \sum_{\substack{k=1\\k\neq j}}^{K} a_k E\{b_j(X_n)b_k(X_m)\}\right]$$

$$= 2\sum_{n=1}^{N} h_{N-n} E\{X_{N+1}b_j(X_n)\}\ , \quad j=1,\ldots,K.$$

IV. EXAMPLES

In this section we consider a particular parametric form for the ZNL and a specific model for the random sequence. The iterative method described earlier is used in this case to determine a filter of the form of Eq.(1). We also determine the mean squared error resulting from use of the optimal filter and that resulting from use of the optimal linear filter. Performances of the filters are compared and it is seen that in several instances the improvement in mean squared error of the suboptimal filter over that of the optimal linear filter is a significant fraction of the corresponding improvement of the optimal filter over that of the optimal linear filter.

Assume that we have knowledge of the regression function

$$r(x) = E\{X_{N+1}|X_N=x\}\ . \tag{6}$$

Notice that if we choose $g(x)=r(x)$ and

$$h_n = \begin{cases}1, & n=0\\0, & n\neq 0\end{cases}\ ,$$

then the estimate would be the same as that of the optimal filter based on the most recent observation. If we were to use the Projection Theorem to choose a different weighting sequence $\{h_n\}$, we might do better. It seems reasonable to expect that if we were to parameterize $g(\cdot)$ in such a way that by proper choice of the parameters we would have $g(x)=r(x)$, and then use this parameterization of the ZNL in the iterative technique described earlier, we might determine a system of the form of Eq.(1) exhibiting very good performance. This is how we will choose the ZNL in this section.

As a model for the random sequence $\{X_n, n=1,2,\ldots\}$ we will assume that

$$X_n = \left(Z_n\right)^{2k+1} \tag{7}$$

where $\{Z_n, n=1,2,\ldots\}$ is a zero mean stationary Gaussian process with unit variance and autocorrelation function $\rho(\cdot)$.

First we will derive an expression for the regression function of Eq.(6) when the random sequence is given by Eq.(7). Using results in [5], we have that

$$E\{X_{N+1}|X_N\} = E\left\{(Z_{N+1})^{2k+1}|Z_n\right\}$$

$$= \sum_{n=0}^{\infty}[\rho(1)]^n\ b_n\theta_n(Z_N)$$

$$= \sum_{n=0}^{\infty}[\rho(1)]^n\ b_n\theta_n\left((X_N)^{1/(2k+1)}\right)\ ,$$

where the series are mean square convergent, the constants $\{b_n\}$ are given by

$$b_n = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} (x)^{2k+1} \theta_n(x) \exp\left(\frac{-x^2}{2}\right) dx \quad , \tag{8}$$

and θ_n is the n-th normalized Hermite polynomial given by

$$\theta_n(x) = \frac{(-1)^n}{\sqrt{n!}} \exp\left(\frac{x^2}{2}\right) \frac{d^n}{dx^n} \exp\left(\frac{-x^2}{2}\right) \quad .$$

We see from Eq.(8) that $b_n=0$ for $n>2k+1$ and, in fact, the b_n's can be obtained from the relation

$$(x)^{2k+1} = \sum_{n=0}^{2k+1} b_n \theta_n(x) \quad .$$

For example, for k=1,

$$b_n = \begin{cases} 3, & n=1 \\ \sqrt{6}, & n=3 \\ 0, & n\neq 1,3 \end{cases}$$

and r(x) is given by $r(x) = [\rho(1)]^3 x + 3\rho(1)\left(1-[\rho(1)]^2\right) x^{1/3}$.
For k=2,

$$b_n = \begin{cases} 15, & n=1 \\ 10\sqrt{6}, & n=3 \\ 2\sqrt{30}, & n=5 \\ 0, & n\neq 1,3,5 \end{cases}$$

and

$$r(x) = [\rho(1)]^5 x + 10[\rho(1)]^3 \left(1-[\rho(1)]^2\right) x^{3/5} + 15\rho(1)\left(1-[\rho(1)]^2\right)^2 x^{1/5} .$$

In general, for an arbitrary positive integer k, it is easily seen that $r(\cdot)$ has the form

$$r(x) = c_{k+1} x + c_k (x)^{(2k-1)/(2k+1)} + c_{k-1}(x)^{(2k-3)/(2k+1)}$$

$$+ \ldots . + c_1 (x)^{1/(2k+1)} \quad ,$$

where the c_i's are constants that can be determined using the above procedure. Thus we choose the ZNL $g(\cdot)$ to be

$$g(x) = \sum_{i=1}^{k+1} a_i (x)^{(2i-1)/(2k+1)}$$

where the parameters a_i are to be determined by the iterative procedure. In utilizing the iterative procedure we encounter the need for the knowledge of moments and joint moments of $\{Z_n\}$ (see [6]), which are given by

$$E\left\{(Z_n)^p\right\} = \begin{cases} 1\cdot 3\cdot 5 \ldots (p-1) & \text{for p even} \\ 0 & \text{for p odd} \end{cases}$$

$$E\left\{(Z_n)^r(Z_{n+i})^s\right\} = \mu(r,s,i) = \begin{cases} (r+s-1)\rho(i)\mu(r-1,s-1,i)+(r-1)(s-1)\cdot \\ \qquad \left(1-[\rho(i)]^2\right)\ \mu(r-2,s-2,i) \text{ when} \\ \qquad (r+s) \text{ is even} \\ 0 \quad \text{when } (r+s) \text{ is odd} \quad . \end{cases} \tag{9}$$

Observing that $\mu(1,1,i)=\rho(i)$ and $\mu(2,2,i)=1+2[\rho(i)]^2$, all higher order joint moments can be calculated using Eq.(9).

In order to compare the performance of the suboptimal estimator with that of the optimal estimator, we have obtained expressions for the mean squared error associated with the optimal estimator. For the optimal system we are interested in

$$E\left\{(Z_{N+1})^{2k+1}|Z_N,\ldots,Z_1\right\} \quad .$$

Notice that this is the (2k+1)-st conditional moment and the conditional distribution has the functional form of a Gaussian distribution. Thus the minimum mean squared error follows using standard properties of the Gaussian distribution (see, for example, [7]). For k=1 we find that the minimum mean squared error is given by $15 - P_1^4\left[9E\{Y^2\} + 6P_1E\{Y^4\} + P_1^2\ E\{Y^6\}\right]$; and for k=2, the minimum mean squared error is given by $945 - P_1^6\ \left[225\ E\{Y^2\} + 300P_1E\{Y^4\} + 130P_1^2E\{Y^6\} + 20P_1^3E\{Y^8\} + P_1^4E\{Y^{10}\}\right]$. In these expressions P_1 is a constant and Y is a normal random variable with zero mean and variance γ^2. The constants P_1 and γ^2 are defined as follows. Assume without loss of generality that the correlation matrix R associated with $Z_1,\ldots,Z_{N+1}$ is positive definite (if it is not, the data can be reduced to achieve this result). Then P_1 is the reciprocal of the element in the lower right corner of R^{-1}. Denote the first N elements in the last row of R^{-1} as $r_1,\ldots,r_N$. Then

$$\gamma^2 = \sum_{i=1}^{N} (r_i)^2 + 2\sum_{m=1}^{N-1}\sum_{n=1}^{m} r_{N-n+1}r_{m-n+1}\rho(N-m) \quad .$$

The mean squared error associated with the optimal linear filter can be obtained in a straightforward fashion.

In the following tables results are presented comparing the suboptimal filter to the optimal filter and the optimal linear filter. Several correlation sequences for $\{Z_n\}$ are considered, both the third power and the fifth power of Z_n are used as models, and examples for two observations and five observations are given. In these tables L_1, L, and L_{min} are the mean squared errors resulting from the optimal linear filter, suboptimal filter using a ZNL, and the optimal filter, respectively. The quantity η_1 is the percent of decrease in L_1 when the suboptimal filter using a ZNL is employed, i.e. $\eta_1 = 100(L_1-L)/L_1$. The quantity η_2 is the percent of possible improvement in L_1 using the optimal filter, i.e. $\eta_2 = 100(L_1-L_{min})/L_1$. The quantity η_3 is the normalized percent of improvement over the linear filter given by the suboptimal filter using a ZNL, i.e. $\eta_3 = 100\ \eta_1/\eta_2 = 100(L_1-L)/(L_1-L_{min})$.

	$\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$	$\rho(5)$
1	.75	.575	.45	.35885	.291
2	.885	.7887	.70762	.639	.5805
3	.55	.315	.187	.11445	.07183
4	.55	.395	.319	.26885	.23023
5	.425	.2375	.14675	.09448	.06207
6	.8333	.6666	.5	.3333	.1666
7	.5787	.2963	.125	.037	.00463
8	.4822	.1975	.0625	.0123	.00077

Table 1. Correlation sequences corresponding to Tables 2-5.

	L_1	L	L_{min}	η_1	η_2	η_3
1	9.1983	8.8614	8.8581	3.6	3.69	97.3
2	5.1744	5.0622	5.0599	2.16	2.21	97.6
3	12.5987	12.1084	12.108	3.89	3.89	99.8
4	12.3196	11.9216	11.8952	3.23	3.44	93.7
5	13.6849	13.2957	13.293	2.84	2.86	99.1
6	6.9247	6.6228	6.4926	4.36	6.23	69.8
7	12.2903	11.732	11.7259	4.54	4.59	98.8
8	13.3219	12.8142	12.8123	3.81	3.82	99.6

Table 2. Mean squared errors and percentages of improvement for k = 1.

	L_1	L	L_{min}	η_1	η_2	η_3
1	727.42	704.58	704.22	3.13	3.18	98.1
2	453.78	444.78	444.49	1.98	2.04	96.7
3	887.49	859.95	859.9	3.1	3.1	99.7
4	879.44	854.59	851.86	2.82	3.13	89.8
5	920.93	899.7	899.43	2.3	2.33	98.5
6	584.57	564.58	550.99	3.41	5.74	59.3
7	876.33	845.86	845.24	3.47	3.54	97.7
8	910.86	884.62	884.42	2.88	2.9	99.2

Table 3. Mean squared errors and percentages of improvement for k = 2.

	h_0	h_1	h_2	h_3	h_4	a_1	a_2
1	.6115	.0127	.008	.0059	.0094	1.519	.6811
2	.7899	.0084	.0064	.0051	.0132	.674	.862
3	.4026	.0093	.0049	.0029	.0024	2.7896	.4114
4	.3654	.0687	.0433	.0297	.028	2.4749	.4334
5	.2827	.0407	.0164	.0076	.0047	3.3779	.2656
6	.776	-.0234	-.0175	-.0111	-.0662	1.2015	.7635
7	.4476	-.028	-.018	-.01	-.0015	2.775	.4375
8	.3505	-.0247	-.0121	-.0032	-.0017	3.342	.3215

Table 4. The coefficients a_i of the nonlinearity $g(x) = a_2 x + a_1 \sqrt[3]{x}$ and the h_i's of the suboptimal system for k = 1.

	h_0	h_1	h_2	h_3	h_4	a_1	a_2	a_3
1	.4779	.0119	.0063	.0042	.0052	4.0527	3.7727	.493
2	.7065	.0097	.0067	.005	.0093	.7136	2.032	.7585
3	.2563	.0059	.0028	.0017	.0014	15.173	4.5019	.196
4	.2466	.0472	.0282	.019	.017	11.733	4.465	.1966
5	.162	.0227	.009	.0043	.0026	23.858	3.802	.0839
6	.6534	-.034	-.0234	-.0136	-.024	2.742	2.9562	.6302
7	.2864	-.0184	-.0096	-.0054	.0002	14.7841	4.5769	.2267
8	.2032	-.0139	-.0065	-.0019	.0008	22.373	4.2663	.128

Table 5. The coefficients a_i of the ZNL $g(x) = a_3x + a_2x^{3/5} + a_1x^{1/5}$ and the h_i's of the suboptimal system for k = 2.

	$\rho(1)$	$\rho(2)$
1	.9	.7
2	.8	.5
3	.8	.3
4	.7	.1

Table 6. Correlation sequences corresponding to Tables 7-8.

	h_0	h_1	a_1	a_2
1	1.2377	-.4974	.9333	.82983
2	.8837	-.3001	1.6639	.6923
3	1.095	-.6467	2.3987	.6089
4	.7927	-.4786	3.2982	.4545

Table 7. The coefficients a_i of the ZNL $g(x) = a_2x + a_1x^{1/3}$ and the h_i's of the suboptimal system for k = 1.

	L_1	L	L_{min}	η_1	η_2	η_3
1	3.7487	3.494	3.1354	6.79	16.3	41.65
2	7.566	7.0273	6.7406	7.12	10.9	65.32
3	5.7804	4.371	1.0231	24.38	82.3	29.62
4	8.9825	7.1689	4.9674	20.19	44.7	45.16

Table 8. Mean squared errors and percentages of improvement for k = 1.

ACKNOWLEDGEMENT

D. Minoo-Hamedani was supported by the Air Force Office of Scientific Research under Grant AFOSR-76-3062 and by the Department of Defense Joint Services Electronics Program under Contract F49620-77-C-0101. G. L. Wise was supported by the Air Force Office of Scientific Research under Grant AFOSR-76-3062. N. C. Gallagher and T. E. McCannon were supported by the Air Force Office of Scientific Research under Grant AFOSR-78-3605.

REFERENCES

1. J.L. Doob, Stochastic Processes, John Wiley, New York, 1953.
2. T. Kailath, ed., Linear Least-Squares Estimation, Dowden, Hutchinson, and Ross, Stroudsburg, Pennsylvania, 1977.
3. I.F. Blake and J.B. Thomas, "On a Class of Processes Arising in Linear Estimation Theory," IEEE Trans. Inform. Theory, Vol. IT-14, pp.12-16, January 1968.
4. A.V. Balakrishnan, "On a Characterization of Processes for which Optimal Mean-Square Systems are of Specified Form," IRE Trans. Inform. Theory, Vol. IT-6, pp.490-500, September 1960.
5. G.L. Wise and J.B. Thomas, "A Characterization of Markov Sequences," Journal of the Franklin Institute, Vol. 299, pp.269-278, April 1975.
6. N.L. Johnson and S. Kotz, Distributions in Statistics: Continuous Multivariate Distributions, p.91, John Wiley, New York, 1972.
7. K.S. Miller, Multidimensional Gaussian Distributions, pp.21-22, John Wiley, New York, 1964.

A NOVEL APPROACH FOR DESIGNING NON-LINEAR DISCRETE TIME FILTERS: PART II

T.E. MCCANNON & N.C. GALLAGHER
School of Electrical Engineering
Purdue University
W. Lafayette, IN 47907

G.L. WISE & D. MINOO-HAMEDANI
Department of Electrical Engineering
University of Texas
Austin, Texas 78712

ABSTRACT

We propose two methods for designing nonlinear discrete time filters. The first method involves an iteration procedure that for simple cases, converges in one or two iterations. However, convergence problems in this approach for higher (> 1) time order filters leads to a second method which is based upon an augmented Hilbert subspace on which the orthogonality principle can be easily applied.

I. INTRODUCTION

In many problems, it can be shown that a non-linear filter either outperforms the linear filter or performs a function not possible with a linear filter. One example of this is the homomorphic processing of speech which utilizes a linear process followed by a non-linearity that is followed by another linear processor [1].

This paper is concerned with the non-linear prediction problem. We consider the system shown in Fig. 1,

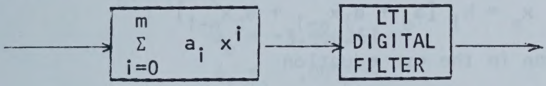


Fig. 1. Non-linear System Under Study.

where we investigate two different methods of design. In Section II, we consider an iterative scheme and give examples of its use. In Section III, we develop a new non-iterative technique motivated by the poor performance of the iterative scheme found in several non-trivial examples. It is also worthwhile to point out that in Part I of this paper, results are obtained based on complete knowledge of the process statistics, while in Part II we only assume that we have a finite sequence of samples from the random process. We also require the random process to be Wide Sense Stationary (WSS) and to have finite higher order moments.

II. ITERATIVE FILTER DESIGN PROCEDURE

We propose the following iterative procedure for determining the MMSE filter coefficients for the system of Fig. 1.

(1) Assume the non-linearity is not present and design the optimum (Wiener) linear filter.
(2) Keeping the unit pulse response of the linear filter constant, compute the polynomial coefficients required to minimize the mse.
(3) With the polynomial coefficients fixed, redesign the optimum linear filter with the polynomial non-linearity.
(4) Repeat Steps (2) and (3) until convergence.

Consider the example where we have a second degree polynomial and a first order linear filter as shown in Fig. 2.

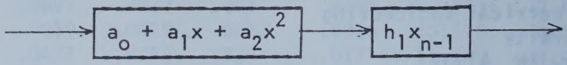


Fig. 2. Example of Non-linear System.

Step 1 tells us to design the optimum linear filter assuming the non-linearity is not present. This linear predictor is given by

$$\hat{x}_n = h_1\, x_{n-1}$$

where $\hat{x}_n \equiv$ estimate of the n^{th} sample of the r.p. x(t)

$x_{n-j} \equiv$ actual value of the $(n-j)^{th}$ sample of x(t)

Using the orthogonality principle

$$E\{(x_n - h_1\, x_{n-1})\, x_{n-1}\} = 0$$

we find the optimum linear filter to be

$$h_1 = \frac{R_x(1)}{R_x(0)} \tag{1}$$

where $R_x(j) \equiv E\{x_n x_{n-j}\}$ and we have made use of the fact that the r.p. is W.S.S.

Step 2 tells us to compute a_o, a_1, and a_2 to minimize the mse keeping h_1 constant. This non-linear predictor is given by

$$\hat{x}_n = h_1\ [a_o + a_1x_{n-1} + a_2x^2_{n-1}] \tag{2}$$

Using this expression in the mse equation

$$\text{mse} = E\{(\hat{x}_n - x_n)^2\}$$

we have that

$$\text{mse} = E\{[h_1(a_o + a_1x_{n-1} + a_2x^2_{n-1}) - x_n]^2\}$$

We minimize the mse with respect to the filter coefficients a_o, a_1, and a_2 by taking partial derivatives

$$\frac{\partial \text{mse}}{\partial a_o} = 0, \quad \frac{\partial \text{mse}}{\partial a_1} = 0, \quad \frac{\partial \text{mse}}{\partial a_2} = 0$$

Then the coefficients a_0, a_1 and a_2 that satisfy the following set of matrix equations are computed:

$$\begin{bmatrix} E\{x_n^2 x_{n-1}\} \\ E\{x_n x_{n-1}\} \\ E\{x_n\} \end{bmatrix} = \begin{bmatrix} h_1\ E\{x^4_{n-1}\} & h_1\ E\{x^3_{n-1}\} & h_1\ E\{x^2_{n-1}\} \\ h_1\ E\{x^3_{n-1}\} & h_1\ E\{x^2_{n-1}\} & h_1\ E\{x_{n-1}\} \\ h_1\ E\{x^2_{n-1}\} & h_1\ E\{x_{n-1}\} & h_1 \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \\ a_0 \end{bmatrix} \tag{3}$$

Step 3 tells us to compute the new optimum linear filter with a_0, a_1 and a_2 constant. The orthogonality principle together with Eq. (2) gives

$$E\{[x_n - h_1\ (a_2x^2_{n-1} + a_1x_{n-1} + a_0)]\ x_{n-1}\} = 0,$$

and we find the new linear filter to be

$$h_1' = \frac{E\{x_n x_{n-1}\}}{a_2\ E\{x_{n-1}^3\} + a_1\ E\{x_{n-1}^2\} + a_o\ E\{x_{n-1}\}} \tag{4}$$

If we solve Eq. (3) for a_o, a_1 and a_2 and substitute these values into Eq. (4) we find

$$h_1' = \frac{R_x(1)}{R_x(0)} = h_1 \tag{5}$$

where the second equality follows from Eq. (1). It is seen that for the non-linear predictor of Fig. 2, the iteration procedure converges in one iteration for an arbitrary W.S.S. r.p. x(t) with finite first, second and third order moments. As an example, consider the following signal

$$x_n = k_1\ x_{n-1}^2 + k_2 + p_1\ u_n$$

where u_n are iid, uniform $(-\frac{1}{2}, \frac{1}{2})$. We can easily obtain the optimum MMSE predictor [2], [3] for this signal by utilizing the conditional expectation

$$\begin{aligned} \hat{x}_n &= E\{x_n | x_{n-1}, x_{n-2}, \ldots\} \\ &= k_1\ x_{n-1}^2 + k_2 \end{aligned} \tag{6}$$

since $E\{u_n | x_{n-1}, x_{n-2}, \ldots\} = 0$. If we let $p_1 = .005$, $k_1 = -1.74$ and $k_2 = 0.87$, the linear filter gives a mse = .293. Calculating the required moments needed for the solution of Eq. (3) empirically with a computer, we find the values of a_o, a_1 and a_2 to be

$$\begin{aligned} a_o &= -3.960261 \\ a_1 &= .005525 \\ a_2 &= 7.913872 \end{aligned}$$

By computer simulation, we find that the non-linear system gives a mse = 2×10^{-6}, a significant improvement over that obtained with the linear filter alone. It is possible to analytically solve for the optimum predictor coefficients; we find the values of a_o, a_1 and a_2 by equating Eqs. (2) and (6) and also using Eq. (1). The values are found to be

$$\begin{aligned} a_o &= -3.960017 \\ a_1 &= 0 \\ a_2 &= 7.920035 \end{aligned}$$

This result agrees very well with the computer simulation.

Next, consider the example as shown in Fig. 3.

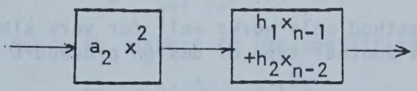


Fig. 3. Example of Non-linear System.

We again apply the iteration procedure as outlined by steps (1), (2) and (3) above. Applying the signal

$$x_n = k_1 x_{n-1}^2 + p_1 u_n \tag{7}$$

where $\{u_n\}$ are iid, uniform $(-\frac{1}{2}, \frac{1}{2})$, we can easily show the procedure terminates after two iterations. However, if we use a general second order polynomial

$$a_2x^2 + a_1x + a_o$$

simulation with the signal of Eq. (7) indicates that convergence is very slow unless the initial choices of a_2, a_1, a_o, h_1 and h_2 are close to the optimum solutions, and then convergence occurs in 2 to 3 iterations. Simulation also shows strong dependence of the final solution on the initial choices of a_2, a_1, a_o, h_1 and h_2. In an attempt to force the solution to the optimum result for the general case, we propose the following modified iteration procedure.

(1) Set $h_1 = h_2 = \cdots = h_k = 1$

(2) Compute the polynomial coefficients required to minimize the mse.

(3) Design the optimum linear filter using the polynomial non-linearity.

(4) Repeat Steps (2) and (3) until convergence.

But even with the modified procedure, simulation indicates sluggish convergence. Fig. 4 demonstrates this convergence with plots of mse versus number of iterations for the polynomial non-linearity $a_2 x^2 + a_1 x + a_o$.

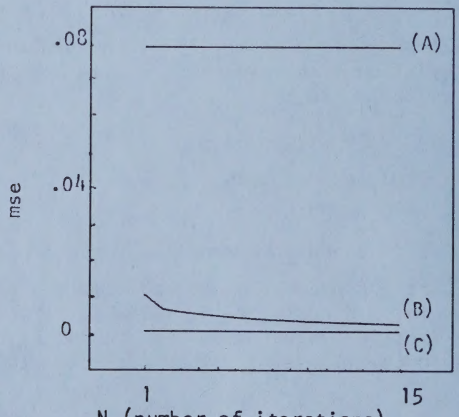
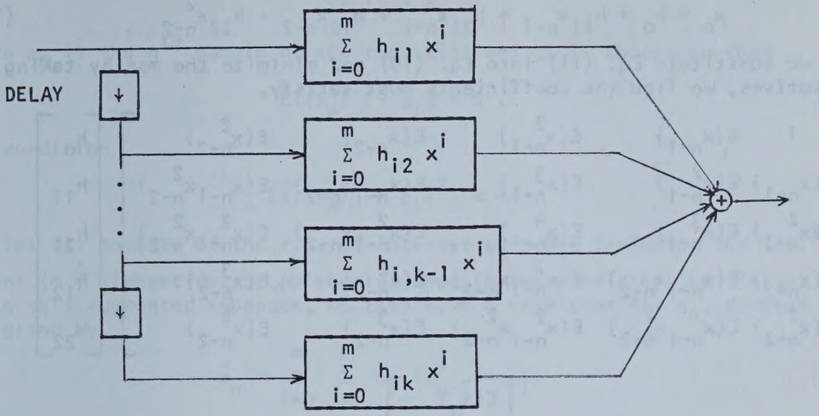


Fig. 4. Demonstration of Convergence. Curve (A) shows mse vs N where the initial values of h_1 and h_2 are the optimum linear filter coefficients. Curve (B) shows mse vs. N for the initial values $h_1 = h_2 = 1$. Curve (C) shows mse vs N where the initial values of h_1 and h_2 are the optimum non-linear coefficients.

It appears that this method only works well for very simple structures and for more general cases another type of design procedure is required.

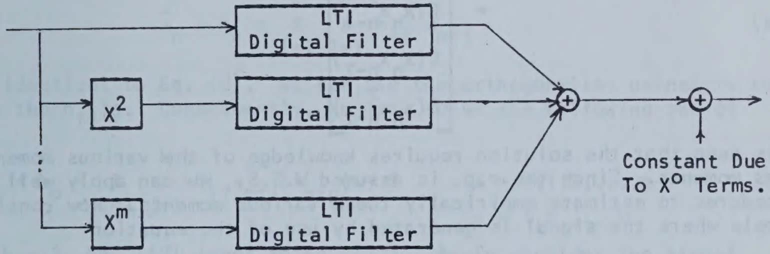
III. NON ITERATIVE FILTER DESIGN METHOD

In Section II, we have studied an iterative procedure for the design of the non-linear predictor in Fig. 1. This system leads to a prediction better than that obtained from the linear predictor, although in many cases the improvement is not significant. We would, however, like to retain the basic structure of Fig. 1, which can also be implemented as shown in Fig. 5 and 6.



DELAY
$\sum_{i=0}^{m} h_{i1} x^i$
$\sum_{i=0}^{m} h_{i2} x^i$
$\sum_{i=0}^{m} h_{i,k-1} x^i$
$\sum_{i=0}^{m} h_{ik} x^i$

Fig. 5. Alternate Structure to the Non-linear System of Fig. 1.



LTI Digital Filter
X^2
LTI Digital Filter
X^m
LTI Digital Filter
Constant Due To X^o Terms.

Fig. 6. Equivalent Structure to the Non-linear System of Fig. 5.

We can now express the non-linear predictor of Fig. 5 as

$$\hat{x}_n = \sum_{j=1}^{k} \left[\sum_{i=0}^{m} h_{ij} x_{n-j}^{i} \right] \tag{8}$$

Defining $h_o \triangleq \sum_{j=1}^{k} h_{oj}$

where the h_{oj} are the constants multiplying the x_{n-j}^{o} terms, we can also write Eq. (8) as

$$\hat{x}_n = h_o + \sum_{j=1}^{k} \sum_{i=1}^{m} h_{ij} x_{n-j}^{i} \tag{9}$$

We now minimize the mse with respect to the coefficients h_{ij} and h_o where

$$\text{mse} = E\{(\hat{x}_n - x_n)^2\} \tag{10}$$

by setting

$$\frac{\partial \text{mse}}{\partial h_o} = 0, \quad \frac{\partial \text{mse}}{\partial h_{ij}} = 0, \quad i = 1,2,\ldots,m \quad j = 1,2,\ldots,k$$

Consider the structure where m = h = 2. Substituting m = h = 2 into Eq. (9) we have the non-linear predictor

$$x_n = h_o + h_{11}x_{n-1} + h_{21}x^2_{n-1} + h_{12}x_{n-2} + h_{22}x^2_{n-2} \tag{11}$$

When we substitute Eq. (11) into Eq. (10) and minimize the mse by taking derivatives, we find the coefficients must satisfy.

$$\begin{bmatrix} 1 & E\{x_{n-1}\} & E\{x^2_{n-1}\} & E\{x_{n-2}\} & E\{x^2_{n-2}\} \\ E\{x_{n-1}\} & E\{x^2_{n-1}\} & E\{x^3_{n-1}\} & E\{x_{n-1}x_{n-2}\} & E\{x_{n-1}x^2_{n-2}\} \\ E\{x^2_{n-1}\} & E\{x^3_{n-1}\} & E\{x^4_{n-1}\} & E\{x^2_{n-1}x_{n-2}\} & E\{x^2_{n-1}x^2_{n-2}\} \\ E\{x_{n-2}\} & E\{x_{n-1}x_{n-2}\} & E\{x^2_{n-1}x_{n-2}\} & E\{x^2_{n-2}\} & E\{x^3_{n-2}\} \\ E\{x^2_{n-2}\} & E\{x_{n-1}x^2_{n-2}\} & E\{x^2_{n-1}x^2_{n-2}\} & E\{x^3_{n-2}\} & E\{x^4_{n-2}\} \end{bmatrix} \begin{bmatrix} h_o \\ h_{11} \\ h_{21} \\ h_{12} \\ h_{22} \end{bmatrix}$$

$$= \begin{bmatrix} E\{x_n\} \\ E\{x_n x_{n-1}\} \\ E\{x_n x^2_{n-1}\} \\ E\{x_n x_{n-2}\} \\ E\{x_n x^2_{n-2}\} \end{bmatrix} \tag{12}$$

It is seen that the solution requires knowledge of the various moments and cross moments. Since the r.p. is assumed W.S.S., we can apply well known procedures to estimate empirically these various moments. Now consider the example where the signal is generated by use of the equation

$$x_n = k_1 x^2_{n-1} + k_2 + p_1\ u_n ;$$

$\{u_n\}$ are iid, uniform $(-\frac{1}{2}, \frac{1}{2})$. From Eq. (6), we know that the optimum predictor is given by

$$\hat{x}_n = k_1 x^2_{n-1} + k_2 \ . \tag{13}$$

This optimum result corresponds to the solution $h_o = k_2$, $h_{21} = k_1$, $h_{11} = h_{12} = h_{22} = 0$.

It is easily shown that this solution satisfies Eq. (12), and hence Eq. (12) leads to the optimum result for the signal in Eq. (13). Likewise, for the general polynomial signal of the form

$$x_n = \sum_{\alpha=0}^{S} \sum_{\beta=0}^{P} \gamma_{\beta\alpha} x^{\beta}_{n-\alpha} + p_1\ u_n \ , \tag{14}$$

Eq. (8) again leads to the optimum solution. These results can also be interpreted in an alternate way. First, define a Hilbert Space over the probability space and the set of r.v. x such that [4], [5]

$$E\{|x|^2\} < \infty$$

with the inner product defined as

$$< x, y > = E\{xy\}$$

We then generate the smallest subspace that contains the elements of the form

$$(x_n)^i - \mu_i$$

where x_n is the n^{th} sample of the r.p. x(t) and μ_i is chosen so that

$$E\{(x_n)^i - \mu_i\} = 0 .$$

The condition

$$E\{|(x_n)^i - \mu_i|^2\} < \infty$$

implies all moments of the r.p. of interest upto and including the $(2m)^{th}$ moment ($m \overset{\Delta}{=}$ highest degree polynomial used in the predictor) be finite. Using this augmented subspace, we then have a predictor for x_n, denoted by $\hat{x}_n$, given by

$$\hat{x}_n = \sum_{i=0}^{m} \sum_{j=1}^{k} h_{ij}(x^i_{n-j} - \mu_i) \tag{15}$$

Expanding Eq. (15) and grouping all the constant terms together, the predictor for x_n becomes

$$\hat{x}_n = h_o + \sum_{i=1}^{m} \sum_{j=1}^{k} h_{ij} x^i_{n-j} \tag{16}$$

which is identical to Eq. (8). We can use the orthogonality principle to determine the h_{ij}'s. Consequently, we must solve the following set of equations

$$E\{(x_n - h_o - \sum_{i=1}^{m} \sum_{j=1}^{k} h_{ij} x^i_{n-j}) x^\ell_{n-p}\} = 0 \qquad \begin{array}{l} \ell = 0,1,2,\dots,m \\ p = 1,2,\dots,k \end{array} \tag{17}$$

When m = h = 2, Eq. (17) leads to Eq. (12). Again consider the signal

$$x_n = -1.74\, x^2_{n-1} + .87 + .005\, u_n ,$$

where $\{u_n\}$ are iid, uniform $(-\frac{1}{2}, \frac{1}{2})$. Simulation for the case k = 1 shows excellent agreement with the known optimum result. However, the determininant of the coefficient matrix vanishes for the case k = 2. This is explained by the observantion that the signal can also be represented as

$$x_n = -1.74\, x^2_{n-1} + .87\, (\alpha+\beta) + .005\, u_n$$

where $\alpha+\beta = 1$. Since

$$x_{n-1} = -1.74\, x^2_{n-2} + .87 + .005\, u_{n-1},$$

then

$$.87\beta = \beta x_{n-1} + 1.74\, \beta x^2_{n-2} - .005\, \beta u_{n-1} ;$$

hence

$$x_n = -1.74\, x^2_{n-1} + .87\alpha + \beta x_{n-1} + 1.74\, \beta x^2_{n-2} + .005\, u_n - .005\, \beta u_{n-1}$$

We note that there are an infinite number of equivalent signal representations and therefore an infinite number of equivalent predictors. This leads to the following design procedure

(1) Set m. (highest desired polynomial degree)

(2) Set k=1. ($k \triangleq$ number of past samples used in the prediction)
(3) Solve for the h_{ij}.
(4) Set k=2. Compute the determinant of the coefficient matrix. If the determinant is zero, terminate; otherwise proceed to step (5).
(5) Solve for the h_{ij}.
(6) Continue incrementing k, either until the determinant vanishes or a desired value of k is reached.

We also note that functions other than polynomials can be used in the predictor. In this case, the predictor is of the form

$$\hat{x}_n = \sum_{i=1}^{m} \sum_{j=1}^{k} h_{ij} \; f_i(x_{n-j}),$$

where we assume the r.v. $f_i(x_{n-j})$ possesses the proper second moment properties. In addition, the $f_i(x)$ should be continuous and bounded over the range of arguments to insure that the augmented subspace is complete and the condition

$$E\{|f_i(x)|^2\} < \infty$$

is satisfied.

IV. SUMMARY AND CONCLUSIONS

In this paper, we investigate two methods of designing non-linear discrete-time filters. The first methods makes use of an iterative procedure, that is alternately computing the linear filter coefficients and the nonlinearity coefficients. We show how this procedure performs by applying it to several examples. Because the resulting filter design is dependent on the initial conditions before iteration, this method is only applicable to certain problems. For example, this procedure appears acceptable when the starting point of the iteration is close to the optimum design. We then present a second non-iterative procedure that makes use of the orthogonality principle over an augmented subspace. The performance of the resulting design is tested by use of several examples and is shown to provide excellent results. This method appears to work well even when the general form of the optimum filter is not known a priori.

REFERENCES

[1] Lawrence R. Rabiner and Bemard Gold, Theory and Application of Digital Signal Processing, Prentice Hall, Englewood Cliffs, New Jersey, p. 689.

[2] Harry L. Van Trees, Detection Estimation and Modulation Theory, Part I, John Wiley and Sons, Inc., New York, New York, pp. 52-74.

[3] Mischa Schwartz and Leonard Shaw, Signal Processing: Discrete Spectral Analysis, Detection and Estimation, McGraw-Hill, pp. 275-314.

[4] Athanasios Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, pp. 385-426.

[5] H. Cramer and M.R. Leadbetter, Stationary and Related Stochastic Processes, John Wiley & Sons, Inc., New York, p. 96.

ACKNOWLEDGEMENT

T.E. McCannon and N.C. Gallagher were supported by the Air Force Office of Scientific Research under grant AFOSR 78-3605. G. L. Wise was supported by the Air Force Office of Scientific Research under grant AFOSR 76-3062. D. Minoo-Hamedoni was supported by the Air Force Office of Scientific Research under grant AFOSR 76-3062 and by the Department of Defense Joint Services Electronics Program under contract F49620-77-C-0101.

DECOMPOSITION ALGORITHMS AND SENSITIVITY ANALYSIS FOR PROBABILISTIC CIRCUITS AND FAULT TREES

ARNON S. ROSENTHAL
University of Michigan, Computer & Communication Sciences
Ann Arbor, Michigan 48109

A decomposition algorithm is devised for finding the unconditional probability that the top gate in a probabilistic single output combinational circuit has output 1 (i.e., "top event" occurs in a system modelled by a fault tree). A second algorithm is given which quickly finds all of the conditional probabilities of the top event (conditioned on each of the circuit-input [basic event] values). The second algorithm is shown to take no more than four times as long as the first, regardless of the number of events.

Both algorithms are based on a new tabular representation for "sufficient" information about a subsystem. This tabular summary gives the probability of each output value set conditioned on each input value set, for that subsystem considered in isolation.

The algorithm for the unconditional probability starts with a summary for each gate or circuit input [basic event] and proceeds by merging pairs of summaries, yielding a summary of the union of the two subsystems. The procedure halts when a summary for the entire circuit is created. Each merge operation is shown to take $O(2^{(\text{# inputs and outputs of both subsystems})})$. If k denotes the largest number of inputs and outputs in any subsystem encountered, the algorithm takes time $O(\text{# events} \cdot 2^k)$. In many practical fault trees, k may be kept quite small. The whole procedure is related to nonserial dynamic programming.

The conditional probability algorithm uses the summaries created by the first algorithm and merges them in such a manner as to obtain summaries of subsystems which include the entire circuit except for one circuit input [basic event]. These summaries give the conditional probabilities.

A number of extensions of the original problem are considered. The algorithm is similar to a method currently being developed for sensitivity analysis in problems solved by nonserial dynamic programming.

A FAULT DETECTING TEST FOR SEMICONDUCTOR RANDOM ACCESS MEMORIES[+]

D. S. SUK[*] and S. M. REDDY
Division of Information Engineering
University of Iowa
Iowa City, Iowa 52242

SUMMARY

Earlier several researchers have designed test procedures for detecting faults in semiconductor random access memories (RAMs) [1-4]. Recently a test procedure requiring 30N operations to detect what have been called functional faults in RAMs was given [4]. In this paper we present a test procedure requiring 16N operations to detect functional faults in N bit RAMs. The proposed test procedure, like the earlier one, belongs to a class of algorithms called march tests. We prove that any march test for the detection of functional faults requires 14N operations and hence the test procedure presented in the paper requires near minimal number of operations.

REFERENCES

1. M. A. Breuer, and A. D. Friedman, "Diagnosis and Reliable Design of Digital Systems", Computer Science Press, Inc., 1976.
2. J. Cocking, "RAM Test Patterns and Test Strategy", Digest of Papers, 1975 Symposium on Semiconductor Memory Testing, pp. 1-8, October 1975.
3. S. M. Thatte and J. A. Abraham, "Testing of Semiconductor Random Access Memories", Proceedings of the Seventh Annual International Conference on Fault-Tolerant Computing, pp. 81-87, June 1977.
4. R. Nair, S. M. Thatte and J. A. Abraham, "Efficient Algorithms for Testing Semiconductor Random-Access Memories", IEEE Trans. Comp., pp. 572-576, June 1978.

[+] The research reported was supported by Air Force Office of Scientific Research under Grant No. AFOSR-78-3582 and by Rome Air Development Center Contract F30602-78-C-0083.

[*] D. S. Suk is currently with Bell Telephone Laboratories, Naperville.

EMPIRICAL COMPUTER SYSTEM DIAGNOSIS*

J. E. SMITH
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706

ABSTRACT

Empirical diagnosis consists of the steps 1) apply tests, 2) replace some unit(s) with a spare, 3) reapply the tests and compare the "before" and "after" test results. The comparison of the two sets of results may indicate that some replaced unit(s) were faulty, and/or it may indicate that further experimentation is necessary in order to arrive at a fault-free system.

This paper first develops a theory of empirical diagnosis and compares empirical diagnosis with the more widely studied predictive diagnosis. The most significant advantage of empirical diagnosis is that it requires less restrictive assumptions, and the main advantage of predictive diagnosis is that it would typically be quicker than the trial-and-error empirical approach. Next, the two strategies are combined and the resulting diagnosis strategy is shown to have several significant advantages.

1. INTRODUCTION

A valuable tool for locating failures in a digital system is the application of diagnostic test sets. Tests are derived for subsystems so that if a subsystem fails, a properly applied test will indicate the failure. Unfortunately, this scheme is more complicated than it first appears because a faulty unit may be used in the application of a test on another unit; i.e. the test may be "invalid".

System diagnosis with test invalidation has been an object of theoretical study for some time [1-5]. Most of this research has centered around models and diagnostic methods of the type first proposed in [1]. Two types of diagnosis methods are commonly studied [1]: one-step diagnosis and sequential diagnosis. In one-step diagnosis, the tests are applied, and, based on the test results, the set of all faulty subsystems are determined and replaced. In sequential diagnosis, the tests are repeatedly applied, and after each test application at least one faulty subsystem is located and replaced. Both of these diagnosis methods have one thing in common: they first involve the application of tests followed by a prediction of faulty units based on the test results. Consequently we call such diagnosis methods predictive.

Predictive diagnosis is based on a number of assumptions that will be discussed in some detail later; an example of such an assumption is the existence of an a priori upper bound, t, on the number of faulty units present. If any of the assumptions are not valid, a predictive diagnosis method is no longer guaranteed to result in a correct system when it finishes; in fact, it may not even be guaranteed to finish. This may be part of the reason that much of the system diagnosis theory thus far developed has apparently not been used in practice.

---

*This work was supported in part by the National Science Foundation under grant ENG78-05778.

In this paper we describe another type of diagnosis method, empirical diagnosis. Here, tests are run, some unit(s) are replaced, and the tests are run again. Then, based on the "before" and "after" test results, it is determined whether a faulty unit was replaced and/or whether further diagnosis is needed. The means for choosing units to be replaced is left open for the moment, but one could make the choice independent of the test results; in this case we say the diagnosis method is purely empirical. A purely empirical diagnosis method is given and shown to result in a correct system under a less restrictive set of assumptions than predictive diagnosis. Nevertheless, purely empirical diagnosis has some practical disadvantages, and these are discussed.

A diagnosis strategy which combines predictive and empirical methods is then given. This "hybrid" strategy has many of the advantages of its constituent parts and few of the disadvantages. The predictive/empirical method is closer to methods actually used in practice, and this paper gives such methods a more solid theoretical foundation.

2. PRELIMINARIES

In [1] Preparata, Metze, and Chien proposed one of the first, and probably the best-known, models for system diagnosis. Since then, several generalizations and extensions have been proposed, most notably in [2-5]. For use in later discussion we now define a diagnosis model similar to the model for morphic systems in [2]. This particular model is chosen mainly because of its generality.

A system S is composed of a set of n potentially faulty units $U = \{u_0, u_1, \ldots, u_{n-1}\}$. There is a set of m tests, $\{t_0, t_1, \ldots, t_{m-1}\}$, that can be applied to the units in S. A test t_j is a complete test for unit u_i if t_j always fails when u_i is the only faulty unit in S and t_j always passes when S contains no faulty units. Each test is assumed to be a complete test for one and only one unit, and at least one complete test exists for each unit. The set of tests that is complete for unit u_i is denoted as $t(u_i)$. For a set of units, $\{u_i, u_j, \ldots, u_k\}$, $t(\{u_i, u_j, \ldots, u_k\}) = t(u_i) \cup t(u_j) \cup \ldots \cup t(u_k)$.

Each test is performed by some set of units. Let $T(u_i)$ be the set of tests applied at least in part by unit u_i. Test invalidation is modelled by assuming that when u_i is faulty the result of a test $t_j \in T(u_i)$ is unreliable in the sense that t_j may pass even though $t_j \in t(u_k)$ and u_k is faulty, or it may fail if u_k is fault-free. $T(\{u_i, u_j, \ldots, u_k\})$ is defined to be $T(u_i) \cup T(u_j) \cup \ldots \cup T(u_k)$; it is this property that makes the system morphic.

The diagnosis model just described can be used to represent a system as a directed graph. There is an internally labeled vertex in the graph for each unit. An edge is directed from the node labeled u_i to the node labeled u_j if $T(u_i) \cap t(u_j) \neq \emptyset$. The edges are labeled with the tests in the set $T(u_i) \cap t(u_j)$. The graphical models of [1,3] result if each test is performed by only one unit.

In system diagnosis, a syndrome is typically defined to be a binary vector representing the test outcomes. We represent a syndrome as an ordered set $\sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_{m-1} \rangle$ where m is the number of tests and σ_i is the outcome of test t_i.

3. PREDICTIVE DIAGNOSIS

Most theoretical results in system diagnosis are based on predictive diagnosis strategies which consist of the steps:

1) apply tests;
2) predict some faulty unit(s) based on the test results;
3) replace the predicted faulty unit(s);
4) for one-step diagnosis, stop;
5) for sequential diagnosis, go to step 1 unless it is predicted that no further units are faulty.

In [1,2] actual diagnosis algorithms are not discussed, just their existence, i.e. by ascertaining the diagnosibility of the systems. In [3] predictive diagnosis algorithms are discussed, as well as results on diagnosibility, and in [6] predictive diagnosis algorithms are the main topic of interest. In the research just mentioned, the following assumptions are either implicitly or explicitly imposed on the diagnosis algorithm.

A1: Faults are permanent.
A2: Tests are complete.
A3: Invalidation information is complete and correct.
A4: Replacements are fault-free and remain fault-free until diagnosis is completed.
A5: There is an a priori upper bound, t, on the number of faulty units initially present.
A6: Only fault-free units are replaced.

In some of the more recent research in the area, some of these assumptions have been removed or relaxed. In [5] faults are allowed to be intermittent. A diagnosis strategy is discussed which consists of repeatedly applying tests and gradually building a syndrome as intermittent faults make their presence known. In [4,7,8], methods are discussed which allow one to raise the upper bound on the number of faults initially present (A5) at the expense of doing away with A6, i.e. fault-free units may be replaced.

As a whole, however, predictive diagnosis algorithms depend on a variety of assumptions which may collectively have a relatively low probability of holding in practice, although the individual assumptions may nearly hold, e.g. 90% fault coverage instead of complete tests. Further, if any of the assumptions fail to hold, a purely predictive method can completely collapse. For example, Fig. 1 shows a diagnostic system that is 1-fault diagnosable [1]. However, if it is assumed that at most one fault is present and two faults are actually present, say u_1 and u_2, then a predictive algorithm will predict u_0 is faulty when $\sigma = <1,1,0>$. Consequently, u_0 is replaced, and since only 1 fault was assumed, a predictive algorithm terminates with a faulty system. Furthermore, if it is determined that the system is still faulty, there is nothing to prevent a predictive algorithm from repeating the same mistake.

4. EMPIRICAL DIAGNOSIS

An alternative to predictive diagnosis is empirical diagnosis. In general, an empirical diagnosis strategy contains the following steps:

1) apply tests, call the syndrome σ^1;
2) replace some unit(s);
3) reapply the tests to get syndrome σ^2. Compare σ^1 and σ^2. The result of this comparison can be used to determine which unit(s) are faulty.

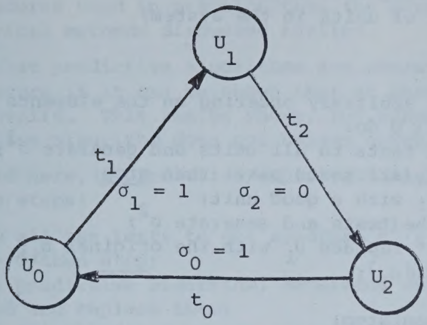


Fig. 1. A diagnostic system with faulty units u_1 and u_2 and syndrome $\sigma = <1,1,0>$.

In an empirical strategy, the steps 1-3 may be repeated. Before giving a more detailed empirical diagnosis algorithm, one additional assumption is given that aids empirical diagnosis.

A7: Test results are reproducible.

That is, if we perform the same tests on the same system (the faults in the system remain unchanged), we get the same test results. This is only necessary at the macrotest level, not at the atomic test level [9]*. In [4,7] a similar assumption is made for a predictive strategy.

We now come to a principal theorem of empirical diagnosis.

Theorem 1: Let σ^1 be the syndrome generated by the first application of tests; let u_i be the replaced unit; let σ^2 be the syndrome generated after u_i is replaced. Then $\sigma^1 \neq \sigma^2$ implies the original u_i was faulty.

Proof: If the original u_i was fault-free, then because of A4 and A7 σ_1 must equal σ_2. □

Definition 1: A set of faulty units U^F is completely detectable if some test will definitely fail if U^F or any nonempty subset of U^F is faulty. See [2] for conditions on detectability.

If a fault is not completely detectable, then at some point in the diagnosis procedure, all the tests might pass despite the presence of faults. When this happens, it appears that any diagnosis algorithm (falsely) concludes that the system is fault-free.

In the empirical diagnosis strategy outlined at the beginning of this section, there was no indication of how units are chosen for replacement in step 2. If this selection is independent of step 1, then we say the strategy is purely empirical. We now give a purely empirical diagnosis algorithm.

---

*Informally, in [9] an atomic test is a single test pattern applied to a subsystem, while a macrotest is a complete test set for the subsystem.

```
EMPDIAG:  an empirical diagnosis algorithm
          $U \leftarrow S$, the set of units in the system;
LOOP:     Do forever;
            $T \leftarrow \emptyset$
            assume some arbitrary ordering on the elements in U;
            for each $u_i \in U$ do:
              apply all tests to all units and generate $\sigma^1$;
              if $\sigma^1 = 0$ (all tests pass) then stop;
              replace $u_i$ with a good unit;
              reapply the tests and generate $\sigma^2$;
              if $\sigma^1 = \sigma^2$ replace $u_i$ with the original $u_i$;
              else $T \leftarrow T \cup \{u_i\}$;
            end;
            if $T = \emptyset$ then stop;
            $U \leftarrow U - T$;
          end LOOP;
```

Example: We apply EMPDIAG to the system in Fig. 1. U initially is $\{u_0 u_1 u_2\}$. The first σ^1 is $<1,1,0>$. Then if we use the subscript ordering, we first replace u_0. Then $\sigma^2 = <1,1,0>$ and we put the old u_0 back in. Next we replace u_1 and the σ^2 that is generated is $<0,1,x>$ (x can be either 0 or 1) and we see that $\sigma^1 \neq \sigma^2$ so $T = \{u_i\}$. Then the new σ^1 becomes $<0,1,x>$ and after u_2 is replaced $\sigma^2 = <0,0,0> \neq \sigma^1$ and T is $\{u_1, u_2\}$. Then, during the second time through the major loop, $\sigma^1 = <0,0,0>$ and the algorithm terminates with a correct system. Only faulty units have been replaced.

> Theorem 2: Given assumptions A1, A2, A4, and A7 algorithm EMPDIAG always terminates with a correct system if a completely detectable fault pattern is initially present.

The only assumption used by EMPDIAG that is not used by most predictive algorithms is A7. In addition, although good units are replaced during EMPDIAG, they are always put back in the system before the algorithm terminates. Hence, at the end of the algorithm A6 is satisfied although during intermediate steps it might not be.

To summarize, EMPDIAG does not require A3 and A5 but does require A7. Although there is no quantitative basis for saying so, it appears that EMPDIAG has a somewhat less restrictive set of assumptions than predictive diagnosis. It has the definite advantage of being able to diagnose any completely detectable fault, which is the best one can hope for.

Unfortunately, purely empirical diagnosis as typified by EMPDIAG has some serious practical flaws. First, since it is basically trial-and-error it may be time-consuming. Second, it requires many exchanges of units which can also be time consuming and in some situations physically difficult. Nevertheless, there may be applications where empirical diagnosis is suitable; for example, where replacement is performed by automatic reconfiguration.

5. PREDICTIVE/EMPIRICAL DIAGNOSIS

In this section we consider empirical diagnosis procedures that are not purely empirical, i.e., the replaced units are selected by considering test results. Consequently, there is a predictive element to the procedure. To look at it another way, we predict faulty units, then run tests again to check to see if the results are consistent with the predictive model and assumptions. In some respects such a procedure is probably closer to diagnostic procedures used in practice than the more theoretical predictive and purely empirical methods discussed earlier.

We assume that predictive algorithms are correct given A1-A6. An inconsistency occurs if it can be shown that at least one of the assumptions A1-A6 must be invalid. This can be shown, for example, by demonstrating that the predictive algorithm does not behave correctly.

As envisaged here, a predictive/empirical diagnosis method consists of the following steps:

```
1) perform all the tests to get syndrome $\sigma^1$;
2) if $\sigma^1 = 0$ then stop;
3) using a predictive algorithm, determine units to be
   replaced and replace them;
4) perform all the tests to get syndrome $\sigma^2$;
5) if $\sigma^1$ and $\sigma^2$ imply an inconsistency then do:
      adjust the model and/or fault prediction algorithm;
      put the units just replaced back into the system
      (optional), and go to 3;
6) else set $\sigma^1 \leftarrow \sigma^2$ and go to 3.
```

Actually, it may be necessary for the "adjustment" mentioned in step 5 to be a call for human intervention.

Some of the empirical consistency checking may be dependent on the particular predictive algorithm used, although a good deal of the checking can be based solely on the model and the assumptions. We show some consistency results of the latter type. The following theorems apply to correct predictive algorithms that are based on the model given in section 2 and the assumptions A1-A6. We begin with a fundamental theorem. σ^1 and σ^2 are as they appear in the above predictive/empirical diagnosis method.

Theorem 3: a) If a predictive algorithm is one-step and $\sigma^2 \neq 0$ then there is an inconsistency.
b) If a predictive algorithm is sequential and $\sigma^1 = \sigma^2 \neq 0$ then there is an inconsistency.

Proof: a) obvious b) If the original u_i were good, then the algorithm would receive the same syndrome, σ^1, as input and would indicate the replacement of a good unit. Hence, the algorithm is not guaranteed to behave correctly and one of A1-A6 must be invalid. □

Theorem 3a) is probably all the inconsistency checking needed for one-step algorithms. For sequential algorithms other checks besides Theorem 3b) are possible, many of which are not as straightforward. In the next theorem, we show further detail about the condition $\sigma_1 \neq \sigma_2$ that is necessary for consistency in a sequential predictive algorithm. We assume that only one unit, u_i, is replaced between the formation of σ^1 and σ^2, although a generalization to multiple replacements is possible.

Theorem 4: Say σ^1 is formed, u_i is replaced, and σ^2 is formed. The replacement of u_i is consistent only if

a) there is a test $t_j \epsilon t(u_i)$ such that $\sigma_j^1 = 1$ and $\sigma_j^2 = 0$, or

b) there is a test $t_k \epsilon T(u_i)$ such that $\sigma_k^1 \neq \sigma_k^2$.

Proof: If u_i were initially faulty, then $\sigma_j^1 = 0$ implies t_j is invalid and could potentially give the same result if u_i were initially fault-free. If for some $t_j \epsilon t(u_i)$, $\sigma_j^2 = 1$, then t_j must be invalid because after replacement u_i is known to be good. Hence if for all tests $t_j \epsilon t(u_i)$ either $\sigma_j^1 = 0$ or $\sigma_j^2 = 1$ then there is no valid test on u_i. Therefore, all these test results could potentially be the same if u_i were initially fault-free.

If for all $t_k \varepsilon T(u_i)$ σ^k does not change, then all of these outcomes would be the same if u_i were initially good. Since u_i does not affect any of the test results not in $t(u_i)$ or $T(u_i)$, then if a) and b) are not satisfied, one could get exactly the same σ^1 if u_i were initially fault-free. This implies the algorithm is not guaranteed to behave correctly, so one of A1-A6 must be invalid. □

In some situations, empirical checking can do more than just find inconsistencies. In some real systems where many units are needed to perform a test, the invalidation information appearing in the model may be incomplete, i.e. A3 does not hold. If A3 is suspect, then empirical checking can be used to determine invalidation information and refine the predictive model.

Theorem 5: Say σ^1 is formed, u_i is replaced, then σ^2 is formed. If A7 holds, then the replacement of u_i is consistent only if for all $t_j \notin T(u_i) \cup t(u_i)$ $\sigma_j^1 = \sigma_j^2$.

If $\sigma_j^1 \neq \sigma_j^2$ then there is an inconsistency that can be explained if u_i invalidates a test not in $T(u_i)$ that is, if u_i invalidates t_j. If incomplete invalidation information is considered likely, then one can adjust the diagnostic model by placing t_j in $T(u_i)$.

The above theorems are far from exhaustive and are just intended to give a flavor of empirical consistency checking that is possible. Further checking can also be done if the predictive algorithm is known.

6. CONCLUSIONS

This paper indicates that procedures for system diagnosis other than the ones mentioned in [1] can be studied in a theoretical fashion. Depending on the cost of the various diagnostic operations and the assumptions used, such alternatives to the predictive procedures may in some cases be superior.

In addition a framework for a higher level diagnosis procedure is shown which includes the predictive algorithms as one of its components. It is also indicated that consistency checking should be another of its components, and this consistency checking can also be studied in a theoretical way — either in conjunction with a particular predictive algorithm or a particular diagnostic model. From this observation, one can conclude that checking for the occurrence of invalid assumptions should be developed as a part of future diagnostic algorithms.

Finally, even if all the assumptions hold, empirical information, i.e. the dynamic behavior of the syndrome as successive units are replaced, might lead to more efficient predictive algorithms than ones which look at a static syndrome and neglect the past history of a sequential diagnosis process.

REFERENCES

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the Connection Assignment Problem of Diagnosable Systems," IEEE Transactions on Electronic Computers, Vol. EC-16, pp. 848-854, Dec. 1967.

[2] J. D. Russell and C. R. Kime, "System Fault Diagnosis: Closure and Diagnosability with Repair," IEEE Transactions on Computers, Vol. C-24, pp. 1078-1089, Nov. 1975.

[3] F. Barsi, F. Grandoni, and P. Maestrini, "A Theory of Diagnosability of Digital Systems," IEEE Transactions on Computers, Vol. C-25, pp. 585-593, June 1976.

[4] A. D. Friedman, "A New Measure of Digital System Diagnosis," Digest of Papers, 1975 International Symposium on Fault-Tolerant Computing, pp. 167-170, June 1975.

[5] S. Mallela and G. M. Masson, "Diagnosable Systems for Intermittent Faults," IEEE Transactions on Computers, Vol. C-27, pp. 560-566, June 1978.

[6] T. Kameda, S. Toida, and F. J. Allan, "A Diagnosing Algorithm for Networks," Information and Control, Vol. 29, pp. 141-148, 1975.

[7] S. Karunanithi and A. D. Friedman, "System Diagnosis with t/s Diagnosability," Proceedings of the Seventh Annual International Conference on Fault-Tolerant Computing, pp. 65-71, June 1977.

[8] J. E. Smith, "Universal System Diagnosis Algorithms," submitted to IEEE Transactions on Computers.

[9] C. R. Kime, "A Theory of Digital System Fault Diagnosis," Tech. Report ECE-77-2, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Feb. 1977.

THE TESTABILITY OF TMR NETWORKS

J. C. SUTTON, III
Michelin Tire Corporation
Sandy Springs, SC 29677

J. G. BREDESON
Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802

SUMMARY

Triple Modular Redundancy (TMR) is used to improve circuit reliability, however, due to the redundancy the networks and the majority gate cannot be tested. If faults occur in more than one of the triplicated circuits then the probability of the output being in error is high. Also, the circuit is only as reliable as the majority gate. If the circuit could be tested during operation, then any faults would be detectable and these errors could be corrected.

If the circuit being triplicated is testable then with additional inputs, gates, and outputs the TMR system will become completely tested. This is illustrated in Figure 1. If the circuit being triplicated is not completely testable, control inputs can be added to the original circuit to make it completely testable [5] and then the scheme of Figure 1 can be used.

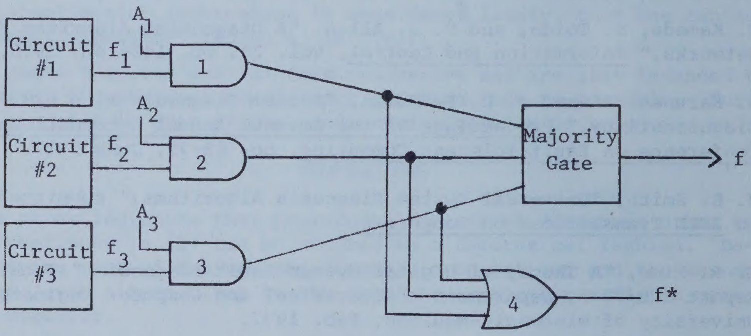


Figure 1. Testable General TMR Network

The control output $f^*=A_1f_1+A_2f_2+A_3f_3$ is irredundant [5] and all lines in the triplicated and test network are therefore testable. The majority gate can be tested by making $f_1=f_2=f_3=1$ and applying all combinations to A_1, A_2, A_3.

REFERENCES

1. Friedman, A. D. and P. R. Memon. Fault Detection in Digital Circuits, Prentice-Hall, Inc., New Jersey, 1971.
2. Abraham, J. A. "A Combinational Solution to the Reliability of Interwoven Redundant Logic Networks," IEEE Trans. on Computers, Vol. C-24, No. 7, pp. 691-695, July 1975.
3. Shertz, D. R. and G. A. Metze. "On the Design of Multiple Fault Diagnosable Networks," IEEE Trans. on Computers, Vol. C-20, No. 11, pp. 1361-1364, November 1971.
4. Poage, J. F. "Derivation of Optimum Tests to Detect Faults in Combinational Circuits," Proc. of the Symposium on Mathematical Theory of Automata, Polytechnic Institute of Brooklyn, pp. 483-528, 1963.
5. Bredeson, J. G. and J. C. Sutton, III. "Fault Detection on Redundant Lines in Multilevel Logic Circuits," Proc. of the 21st Midwest Symposium on Circuits and Systems, pp. 209-214, 1978.

A TESTABILITY MEASURE BASED ON INFORMATION THEORY

J. A. DUSSAULT
Western Electric Company
Engineering Research Center
P. O. Box 900
Princeton, New Jersey 08540

ABSTRACT

Digital circuits are currently being designed with limited emphasis on testability. A testability measure would provide a quantitative measure of the testability of a circuit. This measure, combined with design guidelines, may be used to improve the testability of circuits. The particular testability measure presented in this paper is based on information theory and is applicable to both combinational and sequential circuits. The measure can be computed from experiments, i.e., results of simulation or from a probabilistic treatment of circuit behavior.

INTRODUCTION

As the testing costs become more and more prominent, it is becoming apparent that in order to cut manufacturing costs, digital circuits should contain features that make them easily testable. This, in essence, involves placing more emphasis at the design phase to reduce substantially testing costs and hence reducing the overall life cycle costs.

There are several ways of improving the testability of circuits. One is based on the use of the pleiad of "add-on's" including test points and break points. Another is structured logic. However, many organizations do not have the luxury of an integrated company structure that is needed to support such an approach to logic design. The third approach that we will consider here makes use of testability measures which can be used with the first approach, i.e., a guided first approach, to implement testable design.

There are, at the present time, some methods to compute the testability of digital circuits. The cruder methods are based on simple formulae using gate count, I/O pin count, memory elements count, etc. A more esoteric measure was developed by Stephenson and Grason [5,6]. In his dissertation, Stephenson actually discusses four types of measures and selects one as the best. One of them (not considered the best) uses information entropy. The method presented here is sufficiently different to warrant another look at applying information theory to the definition of a testability measure. It is based on the work by Azema and Lozes

[1,2]. The context and the definitions are somewhat different and other ideas have been incorporated as well.

BASIC CONCEPTS IN INFORMATION THEORY

In information theory, one looks at channels over which are performed experiments that have certain outcomes. The nature of the experiments, the characteristics of the channel and the outcomes are probabilistic in nature. Suppose we have a set of possible outcomes

$$\{a_1, a_2, \ldots, a_i, \ldots, a_k\}$$

then the self information of the event a_i is defined as

$$I(a_i) = \log \frac{1}{P(a_i)} = -\log P(a_i) \quad \text{where } P(a_i) \text{ is the probability of } a_i \text{ occurring.}$$

Intuitively, the argument $1/P(a_i)$ makes sense: if a_i is very likely to occur, then very little information is provided by its occurrence. In fact, if a_i occurs with probability 1, then nothing is learned when it occurs, hence no information is gained. If $P(a_i)$ is close to 0, then the occurrence of a_i is quite rare and quite significant so that the information attached to it is "large." The logarithmic scaling can be justified further: consider that the set $\{a_1, a_2, \ldots, a_k\}$ is a set containing all vectors of length j. There are 2^j such vectors. Assume also that all vectors are equiprobable, i.e., that they can occur with the same probability 2^{-j}. Then the self-information of any given outcome a_i is

$$I(a_i) = -\log 2^{-j} = j \text{ bits}$$

It is intuitively satisfying that j bits of self information are needed to specify a vector of j binary digits, hence the use of logarithms.

In a joint ensemble XY, the information provided about an event $x \varepsilon X$ by the occurrence of an event $y \varepsilon Y$ is called the mutual information $I(x;y)$.

$$I(x;y) = \log \frac{P(x|y)}{P(x)} = I(y;x)$$

Its average is given by

$$I(X;Y) = \sum_x \sum_y P(x,y) \log \frac{P(x|y)}{P(x)}$$

Self information is a special case of mutual information and is a random variable. Its average is called entropy and is defined as follows:

$$H(X) = \sum_x P(x) \log \frac{1}{P(x)} = -\sum_x P(x) \log P(x)$$

This can be extended to conditional entropy which is the average information required to specify $x \varepsilon X$ after $y \varepsilon Y$ is known

$$H(X|Y) = \sum_{x,y} P(x,y) \log P(x|y)$$

Using simple substitutions, the average mutual information can be rewritten as

$I(X;Y) = H(X) - H(X|Y)$ *

Expanding the above $I(X;Y)$ in terms of probabilities, it is easy to verify that $I(X;Y) = I(Y;X)$. Hence

$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ **

$I(X;Y)$, $H(X|Y)$, and $H(Y|X)$ are going to be interpreted as digital circuit characteristics in the following section.

INFORMATION THEORY AND DIGITAL CIRCUITS

In this section we present the intuitive notion of testability and testability measures and relate them to fundamental quantities in information theory.

Definition: Testability is the property of circuits for which efficient test (diagnostic) sets can be obtained efficiently.

This definition essentially involves two time frames: One characterizes the difficulty of producing the test set, and the other refers to its application. Also, the terms efficient refer to both time and effort involved to achieve a predetermined level of fault detection or location. All these notions are intertwined to form the intuitive concept of testability which we will try to define quantitatively in information theory parlance.

Testability is usually broken down into two components that are more tractable: observability and controllability.

Definition: Observability is the ability to determine the internal state of a circuit by looking at its primary outputs.

Definition: Controllability is the ability to control the internal state of a circuit by exercising its primary inputs.

It is intuitive to think that for a given multilevel circuit the components close to the inputs are "easy" to control but "difficult" to observe. Similarly, components close to the outputs are more easily observable than they are controllable. Testability is then some function of observability and controllability.

Definition: A testability (observability, controllability) measure quantizes the testability (observability, controllability) of components and/or circuits.

Clearly, the testability measure should track the intuitive concepts imbedded in the notion testability. Also very important is that the measure should be computationally simple. A very accurate testability measure would be the sum total of all the (weighted) efforts (say in dollars) to generate and apply a test set that meets the desired requirements. The aim of the testability measure as we define it is to predict the testability of the circuit without actually solving the problem of test generation. The testability measure can then be used as a design aid sufficiently inexpensive to permit interactive modification of the circuit to reach an acceptable level of testability.

Let us now define the following measures for a circuit with inputs X and outputs Y:

1) Observability measure: $1/H(X|Y)$
2) Controllability measure: $1/H(Y|X)$
3) Testability measure: $I(X;Y)$

Since $H(X|Y)$ and $H(Y|X)$ are measures of uncertainty, to become measures of information they have either to be compared to another measure of uncertainty or inverted (additive or multiplicative inverse). We chose the multiplicative inverse and kept the comparison approach for the testability measure. The average conditional entropy $H(X|Y)$ gives a measure of the uncertainty about the inputs X given that Y is known. The lesser the uncertainty, the larger the amount of information. If $H(X|Y)$ is large, then there is a large amount of uncertainty about X when the outputs Y are known. Hence, for $H(X|Y)$ large the inputs X cannot be predicted accurately or observed by simply looking at the outputs. Similarly, if it is difficult to predict what the outputs are going to be given the inputs, it can be interpreted as poor controllability. Hence $1/H(Y|X)$ seems appropriate to measure controllability.

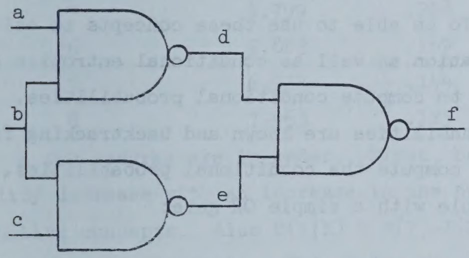
Looking back at equation *, we see that testability is a measure of information obtained by the difference of two uncertainty values. It is also conveniently symmetric. Moreover, $I(X;Y)$ is a function of both $H(X|Y)$ and $H(Y|X)$ as can be seen from both * and **. It is not a straight forward function of observability and controllability, but its intuitive interpretation matches well that of testability.

COMPUTATION OF THE MEASURE

Calculation of the measure is probably the most significant factor in determining its usefulness. Technical purity often has to be compromised in order to make the measure computationally tractable. Typical of the modifications are scaling, weighting, approximations, etc., all of which are designed to keep it simple and to provide a better fit of the intuitive concepts. In this section essentially two approaches are discussed. Both of them are still very close to the theoretical concepts; hence somewhat immature as far as the computational aspects are concerned.

The first approach is "experimental." In this case a set of input vectors is applied to the circuit under consideration (or to its simulation). Statistical records of the outcomes are then used to obtain single and conditional probabilities on lines of interest. Once these are obtained, it is a simple matter to compute the different measures. The experiment may be the application of all possible input vectors or may be the application of sufficiently long random or pseudo-random sequences. This may be objectionable because of the time and accuracy requirements.

Example: The following trivial circuit illustrates the arithmetic involved. The experiment is the application of all possible input vectors.



a b c	d e f
0 0 0	0 0 0
0 0 1	0 0 0
0 1 0	0 0 0
0 1 1	1 0 1
1 0 0	0 0 0
1 0 1	0 0 0
1 1 0	0 1 1
1 1 1	1 1 1

The outcomes are observed at the primary output (f) and on lines d and e. The observability of line d can be computed as follows:

$$H(d|f) = -\sum_{\substack{p=0,1\\ d=0,1}} p(d,f) \log p(d|f)$$

$$= -[5/8 \log 1 + 1/8 \log 1/3 + 0 \log 0^{\dagger} + 1/4 \log 2/3]$$

The observability of line d is then 1/.344 = 2.904 (= observability of line e); similarly,

$$H(c|f) = H(a|f) = -[3/8 \log 3/5 + 1/8 \log 1/3 + 1/4 \log 2/5 + 1/4 \log 2/3]$$

$$= .9512$$

$\therefore$ The observability on lines a and c is 1.0513

$$H(b|f) = -[1/2 \log 4/5 + 0 \log 0 + 1/8 \log 1/5 + 3/8 \log 1] = .4512$$

$\therefore$ The observability of line g = 2.216

$$H(a,b,c;f) = H(a,b,c) - H(a,b,c|f)$$

[†] 0 log 0 is taken as $\lim_{X\to 0^+} X \log X = 0$

The testability of the circuit is given by

$$I\ (a,b,c;f) = H\ (a,b,c) - H\ (a,b,c|f)$$

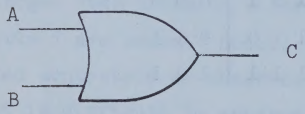
$$= 3 - 2.045 = .954$$

In the following we give an outline of a method whereby it is possible to eliminate the need for an experiment.

Parker and McCluskey have given a probabilistic treatment of general combination networks [3,4]. In their study, they assign probabilities to input signals and give rules for the propagation of probabilities to the outputs.

For an inverter with input A and output B, $P(B=1) = P(A=0) = 1-P(A=1)$, etc. For an AND gate realizing $C = AB$, $P(C=1) = P(A=1) \cdot P(B=1)$. Finally for an OR gate implementing $C = A \vee B$, $P(C=1) = P(A=1) + P(B=1) - P(A=1)P(B=1)$. All these, of course, can be generalized to other gates and to gates with more inputs. To be able to use these concepts to calculate the average mutual information as well as conditional entropies at different points we must be able to compute conditional probabilities. At the beginning, all single probabilities are known and backtracking from the outputs makes it possible to compute the conditional probabilities.

Consider the following example with a simple OR gate:



$$P(C=1) = P(A=1)+P(B=1) - P(A=1)P(B=1)$$

Using basic exioms and theorems of probability we have $P(A=1|C=1)$

$$= \frac{P(A=1 \cap C=1)}{P(C=1)}$$

$$= \frac{P(A=1) + P(C=1) - P(A=1 \cup C=1)}{P(C=1)}$$

However, the event A=1 implies C=1, i.e., $\{A=1\} \subseteq \{C=1\}$, hence

$$P(A=1 \cup C=1) = P(C=1) \quad \text{so that}$$

$$P(A=1|C=1) = \frac{P(A=1)}{P(C=1)}$$

Generalization gives the following:

For an inverter $\bar{A}=B$ $\quad P(A=0|B=1) = 1 \quad P(A=0|B=0) = 0$

$P(A=1|B=0) = 1 \quad P(A=1|B=1) = 0$

For an OR gate with inputs A and output C $\quad P(A=0|C=0) = 1$

$P(A=1|C=0) = 0$

$P(A=1|C=1) = P(A=1)/P(C=1)$

$P(A=0|C=1) = 1-P(A=1)/P(C=1)$

For an AND gate with inputs A and output C

$P(A=0|C=0) = P(A=0)/P(C=0)$

$P(A=1|C=0) = 1-P(A=0)/P(C=0)$

$P(A=0|C=1) = 0$

$P(A=1|C=1) = 1$

Example: Using this approach we compute the different measures for gates that implement function with exactly one "on vertex" or one "off vertex" (e.g., AND, OR, NAND, NOR, etc.). The number of inputs is variable and each input variable is uniformly distributed and assumed independent from any others.

#inputs = H(X)	H(X|Y)	1/H(X|Y)	I(X;Y)-H(Y)
2	1.188	.841	.811
3	2.456	.407	.543
4	3.662	.273	.337
5	4.799	.208	.200
6	5.883	.169	.116
7	6.934	.144	.066
8	7.963	.125	.037

A few remarks are in order. First, both observability and testability decrease with an increase in the number of inputs reflecting the intuitive concepts. Also $H(Y|X) = H(Y)-I(X;Y)$ is zero throughout giving infinite controllability. This holds for irredundant combinational circuits, and whether infinite controllability is acceptable needs to be investigated further. It is easy to compute conditional probabilities where there are few levels. However, with several levels it becomes very difficult to compute those probabilities; and this is another area where further analysis is needed.

CONCLUDING REMARKS

Azéma and Lozes have discussed the placement of test points using information theory for circuits that are to be subjected to probabilistic testing [2]. They use points with low $I(X;Y)$ as test points candidates. We suggest points with low observability for output test points and lead cutting and input test point insertion at locations with low controllability values. Therefore, in view of the present discussion, points with high $H(X|Y)$ and $H(Y|X)$ appear to be good candidates for test points.

As mentioned earlier, many of the ideas discussed in this paper need further development. Among the more important are the computational aspects and the calibration of the measures.

REFERENCES

1. Azéma, P. and Lozes, A., "Insertion de points de Test" internal report, L.A.A.S., C.N.R.S., France, December 1974.

2. Lozes, A., "Evaluation et accroissement de la testabilite des systemes logiques," These de doctorat, Universite Paul Sabatier, Toulouse, 1975.

3. Parker, K. P. and McCluskey, E. J., "Analysis of Logic Circuits with Faults Using Input Signal Probabilities," IEEE Transaction on Computers, Vol. C-24, pp. 541-542, May 1975.

4. Parker, K. P. and McCluskey, E. J., "Probabilistic Treatment of General Combinational Networks," IEEE Transaction on Computers, Vol. 24, pp. 668-670, June 1975.

5. Stephenson, J., "A Testability Measure for Register-Transfer Level Digital Circuits," Ph.D. Dissertation, Department of Electrical Engineering, Carnegie-Mellon University, Pittsburg, November 1974.

6. Stephenson, J. E. and Grason, J., "A Testability Measure for Register Transfer Level Digital Circuits," Proceedings of the 6th Symposium on Fault Tolerant Computing, pp. 101-107, June 1976.

ROVING DIAGNOSIS: DETERMINISTIC ANALYSIS OF A SCHEME FOR SYSTEM SELF-DIAGNOSIS+

RAVINDRA NAIR, GERNOT METZE, and JACOB ABRAHAM
Coordinated Science Laboratory
University of Illinois, Urbana, Illinois 61801

ABSTRACT*

This paper studies the design of systems capable of continuous self-diagnosis by using a principle called "roving diagnosis". In essence, roving diagnosis requires one part of the system to be diagnosing another while the remainder of the system continues normal operation. The part of the system most recently diagnosed as fault-free now takes its turn to diagnose another part. Thus there is a subsystem involving the diagnosed and diagnosing units which apparently roves through the system until no part of the system remains undiagnosed. Thus there is no need for a global mechanism. The system is well utilized and only a part of the system has to shut down during diagnosis.

The basic model for the system which will be used in this paper is the system diagnostic graph as described by Russell and Kime [1]. In general, there may be more than one unit required to test any given unit. Hence by looking at the diagnostic graph for the system it is not always immediately obvious whether diagnosis by roving is possible. It can be shown however, that if the diagnostic graph for the system contains a particular kind of graph, called a roving graph, then the system can perform roving diagnosis.

The success of the roving diagnosis scheme depends to some extent on the ability to test certain distinguished nodes in the system, identifiable from the roving graph, called the initial nodes. This paper describes algorithms for minimizing the number of initial nodes in certain types of systems. Various schemes by which the initial nodes themselves may be tested, with almost no hard-core requirement, will be evaluated.

Associated with the problem of locating faulty units in systems is the problem of reconfiguring the system around the faults so that the system may continue operation possibly with a degradation in performance. All the nodes which were formerly communicating with the faulty node have to be informed about the existence of the fault and the system should continue useful operation once the faulty node has been effectively removed.

The paper will address these problems and show their relation to some known graph-theoretic problems. A detailed treatment can be found in [2].

REFERENCES

[1] J.D. Russell and C.R. Kime, "System fault diagnosis: closure and diagnosability with repair," IEEE Trans. Comput., vol. C-24, pp. 1079-1088, Nov. 1975.

[2] R. Nair, "Diagnosis, self-diagnosis and roving diagnosis in distributed digital systems," Coord, Sci. Laboratory, Univ. of Illinois, Urbana, Report R-823, Sept. 1978.

+This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract No. DAAB-07-72-C-0259.

*Since the authors are from the University of Illinois, only an abstract is published.

ON THE DETECTION OF FAULT-FREE UNITS IN A SYSTEM+

RAVINDRA NAIR, GERNOT METZE, and JACOB ABRAHAM
Coordinated Science Laboratory
University of Illinois, Urbana, Illinois 61801

ABSTRACT*

Research in the area of diagnosis of systems was launched by Preparata, Metze, and Chien [1]. Their work, the connection assignment problem, considered the case when one subsystem can completely test another. A more general case was considered by Russell and Kime [2,3]. A system is defined to be d-fault diagnosable with repair if and only if there exists a sequence of applications of a test set and repairs of identified faults that allow all faults (at most d in number) originally present to be identified. In certain applications, as in airline reservation systems, it may not be desirable to have any faulty units around, so that the integrity of the data base can be maintained. Under such circumstances, the only acceptable solution may involve the identification of all faults on one application of the test sequence (called d-fault diagnosability without repair). This condition has been shown to be very demanding in terms of nodes and interconnections required.

This paper studies an alternative approach to the problem. Instead of designing systems in which at least one faulty unit can be identified after an application of the test sequence, one could design for systems in which at least one fault-free unit can be identified after an application of the test sequence. (The fault-free units could then proceed to execute tasks vital to the integrity of the system, while further diagnosis is continued on that part of the system known to contain the fault.)

Necessary and sufficient conditions have been derived for finding a fault-free unit in a general system. Interesting results have been obtained for some specific classes of systems. For example, a single-mask-per-test (SMPT) system is one in which every test for a fault is invalidated by (i.e. gives unreliable results in the presence of) at most one other fault in the system. For such systems where one unit can be completely tested by another, the problem of finding one good unit is never more difficult than the problems of finding one bad unit. A detailed treatment is given in [4].

REFERENCES

[1] F.P. Preparata, G. Metze, and R.T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. Electron. Comput., vol. EC-16, pp. 848-854, Dec. 1967.

[2] J.D. Russell and C.R. Kime, "System fault diagnosis: closure and diagnosability with repair," IEEE Trans. Comput., vol. C-24, pp. 1079-1088, Nov. 1975.

[3] J.D. Russell and C.R. Kime, "System fault diagnosis: masking, exposure and diagnosability without repair," IEEE Trans. Comput., vol. C-24, pp. 1155-1161, Dec. 1975.

[4] R. Nair, "Diagnosis, self-diagnosis and roving diagnosis in distributed digital systems," Coord. Sci. Laboratory, Univ. of Illinois, Urbana, Report R-823, Sept. 1978.

+This work was supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract No. DAAB-07-72-C-0259.

*Since the authors are from the University of Illinois, only an abstract is published.

INVARIANCE AND FEEDBACK FOR FINITE-STATE MACHINES*

P.E. LIEPA and W.M. WONHAM
Dept. of Electrical Engineering
University of Toronto
Toronto, Canada M5S 1A4

ABSTRACT

Concepts of 'invariance under a map' for subsets and partitions are extended to more general invariance concepts corresponding to various systems properties. Extremal invariant objects are obtained as fixpoints of corresponding lattice mappings. In this framework, the disturbance decoupling theory of [1] is extended into the setting of finite-state machines.

1. PRELIMINARIES

1.01 Notation

$x := y$ means 'x is defined as y'. $\mathbb{N} := \{0,1,2,\ldots\}$. A function $f:X \to Y:x \longmapsto f(x)$ is variously abbreviated as $f:X \to Y$, $x \longmapsto f(x)$, and f.

1.02 Definition

Let X be a finite set. A partition on X is an onto map $\theta:X \to X/\theta$ projecting each $x \in X$ to the block of θ containing it. There is a corresponding equivalence relation:

$$x \equiv y\ [\theta] \quad \text{iff} \quad \theta(x) = \theta(y)\ .$$

Let $S(X)$:= the set of all subsets of X

$C(X)$:= the set of all partitions on X .

1.03 Definition

The reader is referred to [4] for the elementary definition lattice theory. A lattice L is complete whenever $\sup(K)$ and $\inf(K)$ exist for all $K \in S(L)$. Define $\bot_L := \inf(L)$ and $\top_L := \sup(L)$.

1.04 Examples

$S(X)$ and $C(X)$ are complete lattices. In $S(X)$, the ordering relation is $\subset$, and the meet and join operations are respectively $\cap$ and $\cup$. The supremal element of $S(X)$ is $\top_{S(X)} = X$, and the infimal element is $\bot_{S(X)} = \emptyset$. The ordering relation in $C(X)$ is defined as follows. For $\sigma,\tau \in C(X)$, $\sigma \leq \tau$ iff, for $x,y \in X$, $x \equiv y\ [\sigma]$ implies $x \equiv y\ [\tau]$. The meet and join operations in $C(X)$

---

[*] This work was supported in part by the National Research Council of Canada, through a Postgraduate Scholarship (P.E.L.), and Operating Grant A-7399 (W.M.W.). The authors are with the Systems Control Group, Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada M5S 1A4.

are denoted by $\wedge$ and $\vee$ respectively. For $\Theta \subset C(X)$,

(1.041) $x \equiv y \; [\inf(\Theta)]$ iff $x \equiv y \; [\theta]$ for all $\theta \in \Theta$

and

(1.042) $x \equiv y \; [\sup(\Theta)]$ iff there exist $n \in \mathbb{N}$ and a sequence $x = z_0, z_1, \ldots, z_n = y$ such that $z_{i-1} \equiv z_i \; [\theta_i]$, $i = 1, \ldots, n$, where each $\theta_i \in \Theta$.

Also, $x \equiv y \; [\bot_{C(X)}]$ iff $x = y$, and $x \equiv y \; [\top_{C(X)}]$ for all $x, y \in X$.

1.05 Definition

Let L be a lattice. Any totally ordered subset of L is called a *chain*. A mapping $f: L \to L$ is *isotone* if $x \le y$ implies $f(x) \le f(y)$, and is *continuous* if f preserves sups and infs of chains. A *fixpoint* of f is any $x \in L$ such that $f(x) = x$.

1.06 Theorem

[Tarski] If L is complete and $f: L \to L$ is continuous then the set P of fixpoints of f is nonempty. Indeed

$$\inf(P) = \inf\{x \in L \mid x \ge f(x)\} \in P$$

and

$$\sup(P) = \sup\{x \in L \mid x \le f(x)\} \in P.$$

1.07 Algorithm

Suppose L and f are as in 1.06. Let $x_0 := \bot_L$, $x_{i+1} := f(x_i)$, $i \in \mathbb{N}$, and set $f^{\uparrow} := \sup(x_i)$. Then $f^{\uparrow} = \inf(P)$. Let $y_0 := \top_L$, $y_{i+1} := f(y_i)$, $i \in \mathbb{N}$, and set $f^{\downarrow} := \inf(y_i)$. Then $f^{\downarrow} = \sup(P)$.

1.08 Example

Consider the discrete time linear system

$$x' = Ax + Bu \; ; \qquad y = Cx , \tag{1.09}$$

where u,x,y belong [resp.] to the input, state, and output spaces [resp.] $\mathcal{U}, \mathcal{X}, \mathcal{Y}$, and $A: \mathcal{X} \to \mathcal{X}$, $B: \mathcal{U} \to \mathcal{X}$, and $C: \mathcal{X} \to \mathcal{Y}$. Let $\mathcal{B} := \operatorname{Im}(B)$. Then the controllable subspace [1] is

$$\begin{aligned} \langle A \mid \mathcal{B} \rangle &= \inf\{\mathcal{R} \subset \mathcal{X} \mid \mathcal{R} \supset A\mathcal{R} + \mathcal{B}\} \\ &= (\mathcal{R} \longrightarrow A\mathcal{R} + \mathcal{B})^{\uparrow} \qquad \text{(by 1.07)} \\ &= \mathcal{B} + A\mathcal{B} + A^2\mathcal{B} + \ldots \end{aligned}$$

We can compute the subspace [1, Chapter 4]

$$\mathcal{V}^* = \sup\{\mathcal{V} \subset \mathcal{X} \mid \mathcal{V} \subset \mathcal{K} \cap A^{-1}(\mathcal{V} + \mathcal{B})\} \qquad \text{(for arbitrary } \mathcal{K} \subset \mathcal{X}\text{)}$$

and the 'unobservable subspace'

$$\mathcal{N} = \sup\{\mathcal{M} \subset \mathcal{X} \mid \mathcal{M} \subset \operatorname{Ker}(C) \cap A^{-1}\mathcal{M}\}$$

analogously using Algorithm 1.07.

1.10 Remark

The lattice functions used in the sequel are all continuous, allowing us to use Theorem 1.06 and Algorithm 1.07 freely throughout.

2. SUBSETS AND INVARIANCE

2.01

Let $\alpha : X \to X$ be a map. A subset $W \subset X$ is α-invariant if $\alpha(W) \subset W$. Now consider the simple system

$$x' = \alpha(x)\ , \qquad x \in X := \text{state set}\ . \tag{2.02}$$

We want our state always to be in a given $K \subset X$. The largest set of initial states that allows this is

$$\begin{aligned} &\sup\{W \mid W \subset K \text{ and } \alpha(W) \subset W\} \\ &= \sup\{W \mid W \subset K \cap \alpha^{-1}(W)\} \\ &= (W \longmapsto K \cap \alpha^{-1}(W))^{\downarrow} \qquad \text{(by 1.07)}\ . \end{aligned}$$

This set is the largest α-invariant subset of K.

2.03

We generalize the idea of 2.01 to the finite-state machine

$$x' = \delta(u,x)\ , \tag{2.04}$$

where $\delta : U \times X \to X$ and $u \in U :=$ input set.

Introduce the subset mappings

$$\overset{\rightarrow}{\delta} : W \longmapsto \delta(U,x)$$

and

$$\delta^{\forall} : W \longmapsto \{x \mid \delta(U,x) \subset W\}\ .$$

A trajectory starting in the subset

$$\begin{aligned} W^* &= \sup\{W \mid W \subset K \text{ and } \overset{\rightarrow}{\delta}(W) \subset W\} \\ &= \sup\{W \mid W \subset K \cap \delta^{\forall}(W)\} \\ &= (W \longmapsto K \cap \delta^{\forall}(W))^{\downarrow} \end{aligned}$$

will always stay there (and hence in K) regardless of input. Further, a trajectory starting in

$$(W \longmapsto W^* \cup \delta^{\forall}(W))^{\uparrow} \tag{2.041}$$

will be in K for all but a finite number of steps.

By analogy with well known concepts in topological dynamics, W^* might be called a stable set or attractor, and (2.041) its basin, or range of attraction.

2.05 Feedback

Can we keep the state of system (2.04) in K using a feedback $\phi : X \to U$? Let $\delta_\phi : x \longmapsto \delta(\phi(x), x)$ and define a new subset map

$$\delta^{\leftarrow} : W \longmapsto \{x \mid \delta(U,x) \cap W = \emptyset\}$$

Then $V \subset \delta^{\leftarrow}(V)$ iff there exists a $\phi : X \to U$ such that $\delta_\phi(V) \subset (V)$. Hence, if $V^* := (V \longmapsto K \cap \delta^{\leftarrow}(V))^{\downarrow}$, there exists a feedback keeping the system trajectories in V^* whenever they start there.

2.06 The Disturbance Containment Problem (DCP)

Add a 'disturbance set' Q and augment the transition map:

$$x' = \varepsilon(q, \delta(u,x)) , \qquad (2.07)$$

$q \in Q$, $\varepsilon : Q \times X \to X$. Define $\varepsilon_\phi : Q \times X \to X : (q,x) \longmapsto \varepsilon(q, \delta_\phi(x))$. Given a subset K of X, the DCP is to find (if possible) a non-empty set $V \subset K$ such that for some feedback $\phi : X \to U$,

$$\varepsilon^{\rightarrow}_{\phi}(V) \subset V .$$

2.08 Theorem

V solves the DCP iff $V \subset \Lambda(V)$, where $\Lambda : S(X) \to S(X) : W \longmapsto K \cap \delta^{\leftarrow}(\varepsilon^{\forall}(W))$.

Proof. Suppose that V solves the DCP. Then

$$\varepsilon^{\rightarrow}_{\phi}(V) \subset V \qquad (2.081)$$

and

$$V \subset K \qquad (2.082)$$

Noting that

$$\varepsilon^{\rightarrow}_{\phi}(V) = \varepsilon^{\rightarrow}\delta_{\phi}(V) , \qquad (2.083)$$

we have that

$$\begin{aligned} V &\subset \delta^{\leftarrow}\delta_{\phi}(V) \\ &\subset \delta^{\leftarrow}\varepsilon^{\forall}\varepsilon^{\rightarrow}\delta_{\phi}(V) \\ &= \delta^{\leftarrow}\varepsilon^{\forall}\varepsilon^{\rightarrow}_{\phi}(V) && \text{(by 2.083)} \\ &\subset \delta^{\leftarrow}\varepsilon^{\forall}(V) && \text{(by 2.081)} \qquad (2.084) \end{aligned}$$

Thus, by (2.082) and (2.084), we have that $V \subset \Lambda(V)$.

Conversely, if $V \subset K \cap \delta^{\leftarrow}\varepsilon^{\forall}(V)$, then $V \subset K$, and $V \subset \delta^{\leftarrow}\varepsilon^{\forall}(V)$ implies (cf. 2.05) the existence of a $\phi : X \to U$ such that $\delta_\phi(V) \subset \varepsilon^{\forall}(V)$. If follows that

$\vec{\varepsilon}_{\phi}(V) = \vec{\varepsilon}\delta_{\phi}(V) \subset \vec{\varepsilon}\varepsilon^{\forall}(V) \subset V$, as required.

The following gives a constructive criterion for solvability of the DCP.

2.09 Corollary

The DCP is solvable iff $\Lambda^{\downarrow} \neq \emptyset$.

2.10 Applications

In passing, we note that a continuous time variant of the W^* problem in 2.03 appears in the population biology literature [5]. Also, the DCP appears quite naturally in game situations, where an opposing player's moves can be regarded as disturbance inputs [3].

3. PARTITIONS AND INVARIANCE

3.01 Notation

Systemically, we can interpret the elements of $C(X)$ as 'output' maps. Denote by (n.nn+θ) the system (n.nn) with output $y = \theta(x)$, $\theta\in C(X)$. We use X/θ to denote the set of blocks of θ. As such, $X/\theta \subset S(X)$.

3.02 Duality

Although we will not dwell on the category theoretical duality between subsets and partition, this duality motivates the translation of several $S(X)$ concepts from §2 into $C(X)$ concepts in this section. A partition $\theta\in C(X)$ is α-invariant (cf. 2.01) if $\theta \leq \theta\circ\alpha$ (equivalently, $x \equiv y\ [\theta]$ implies $\alpha(x) \equiv \alpha(y)\ [\theta]$). In the sequel, we shall use $\gamma\in C(X)$ as the $C(X)$-counterpart of the subset $K\in S(X)$ used in §2. Then the $C(X)$-counterpart of $(W \longmapsto K\cap\alpha^{-1}(W))^{\downarrow}$ is $(\theta \longmapsto \gamma\wedge(\theta\circ\alpha))^{\downarrow} = \inf_{n\in\mathbb{N}}(\gamma\circ\alpha^{n})$, the 'observer' (as introduced in [2]) for system (2.02+γ).

3.03 Definition

Define the partition mapping

$$\vec{\delta}: C(X) \to C(X): \theta \longmapsto \inf\{\sigma\in C(x) \mid \text{ for all } u,v\in U: \\ x\equiv y\ [\theta] \text{ implies } \delta(u,x)\equiv\delta(v,y)\ [\sigma]\}.$$

This mapping and its homograph (defined in 2.03) will always be distinguished by the arguments they take. This will also apply to partition mappings defined later on. A partition $\theta\in C(X)$ is input invariant iff $\theta \geq \vec{\delta}(\theta)$. For the sequence $\bar{u}\in U^{\mathbb{N}}$, denote by $x(\bar{u},t)$ the state at time $t\in\mathbb{N}$ of the trajectory of (2.04) starting at x and induced by $\bar{u}$. The system (2.04+θ) is input decoupled iff

$$x(\bar{u}_1,t) \equiv x(\bar{u}_2,t)\ [\theta]$$

for all $x\in X$, $t\in\mathbb{N}$, and all $\bar{u}_1,\bar{u}_2\in U^{\mathbb{N}}$. Define $\rho(\delta) := (\vec{\delta})^{\uparrow} = \inf\{\theta \mid \theta$ is input invariant$\}$. Then

3.04 Theorem

(2.04+γ) is input decoupled iff $\rho(\delta) \leq \gamma$.

Proof. Define

$$\mu := \inf\{\theta \in C(X) \mid (2.04+\theta) \text{ is input decoupled}\} ,$$

and $\rho := \rho(\delta)$. The theorem is equivalent to the assertion that $\mu = \rho$. Clearly (2.04+ρ) is input decoupled, so $\mu \leq \rho$.

To show that $\mu \geq \rho$, define

$$\mu(x,t) := \inf\{\theta \in C(X) \mid x(\bar{u}_1,t) \equiv x(\bar{u}_2,t)\ [\theta] \text{ for all } \bar{u}_1,\bar{u}_2 \in U^{\mathbb{N}}\}.$$

Then

$$\mu = \sup\{\mu(x,t) \mid (x,t) \in X \times \mathbb{N}\} . \tag{3.041}$$

The reader may verify that, for any $(x,t) \in X\times\mathbb{N}$, and $y \neq z$:

(3.042) $y \equiv z\ [\mu(x,t)]$ iff there exist $\bar{u}_1,\bar{u}_2 \in U^{\mathbb{N}}$ such that $x(\bar{u}_1,t) = y$ and $x(\bar{u}_2,t) = z$ and, as a corollary,

(3.043) $y \equiv z\ [\mu(x,t)]$ implies that

$$\delta(u,x) \equiv \delta(v,y)\ [\mu(x,t+1)] \text{ for all } u,v \in U.$$

To show that $\mu \geq \rho$, it suffices to show, by definition of ρ, that μ is input invariant. Suppose $x \equiv y\ [\mu]$. Then, by (3.041) and (1.042), there exist $n \in \mathbb{N}$ and sequences $z_0,\ldots,z_n \in X$ and $(x_1,t_1),\ldots,(x_n,t_n) \in X\times\mathbb{N}$ such that $x = z_0$, $y = z_n$ and, for $i = 1,\ldots,n$, $z_{i-1} \equiv z_i\ [\mu(x_i,t_i)]$ and $z_{i-1} \neq z_i$. Now, if for any $u,v \in U$ we can show that $\delta(u,x) \equiv \delta(v,y)\ [\mu]$, we are done. Let $z_0' = \delta(u,z_0)$, and $z_i' = \delta(v,z_i)$, $i = 1,\ldots,n$. Then, by (3.043), $z_{i-1}' \equiv z_i'\ [\mu(x_i, t_i+1)]$ for $i = 1,\ldots,n$. Noting that $z_0' = \delta(u,x)$ and $z_n' = \delta(v,y)$, we conclude that $\delta(u,x) \equiv \delta(v,y)\ [\mu]$, as required.

3.05 Example

For the linear system (1.09), let $P : \mathcal{X} \to \mathcal{X}/\langle A|\mathcal{B}\rangle$ be the canonical projection. Then $P = \rho(x \longmapsto Ax+Bu)$. The following statements are equivalent:

(i) $P \leq C$ (considering P and C as partitions)

(ii) There exists a map $D : \mathcal{X}/\langle A|\mathcal{B}\rangle \to \mathcal{Y}$ such that $C = DP$

(iii) $\langle A|\mathcal{B}\rangle \subset \ker C$.

We conclude from Theorem 3.04 that (1.09) is input decoupled iff $\langle A|\mathcal{B}\rangle \subset \ker C$.

3.06 The Disturbance Decoupling Problem (DDP)

For (2.07+γ) and $\phi : X \to U$ we recall that

$$\varepsilon_\phi : (q,x) \longmapsto \varepsilon(q,\delta_\phi(x)) .$$

The DDP is to find (if possible) a feedback ϕ such that

$$\rho(\varepsilon_\phi)\le\gamma\,.$$

By Theorem 3.04, if such a feedback is applied, the output of (2.07+γ) will be independent of the disturbance inputs.

3.07 Lemma

The feedback $\phi:X\to U$ solves the DDP for (2.07+γ) iff there exists a partition $\nu\in C(X)$ such that $\overrightarrow{\varepsilon}_\phi(\nu)\le\nu\le\gamma$.

Proof. Suppose ϕ solves the DDP. Then $\rho(\overrightarrow{\varepsilon}_\phi)=(\overrightarrow{\varepsilon}_\phi)^{\uparrow}$ is the required ν. Conversely, suppose ν satisfies $\overrightarrow{\varepsilon}_\phi(\nu)\le\nu$ and $\nu\le\gamma$. By definition,

$$\rho(\overrightarrow{\varepsilon}_\phi)=\inf\{\theta\in C(X)\,|\,\overrightarrow{\varepsilon}_\phi(\theta)\le\theta\}\,,$$

so $\nu\ge\rho(\overrightarrow{\varepsilon}_\phi)$, and hence $\rho(\overrightarrow{\varepsilon}_\phi)\le\gamma$.

3.08 Definition

We define $\overleftarrow{\delta}:C(X)\to C(X)$ (distinguished from $\overleftarrow{\delta}:S(X)\to S(X)$ by its argument) as follows. For a partition $\theta\in C(X)$,

$$x\equiv y\ [\overleftarrow{\delta}(\theta)]\ \text{iff, for all}\ b\in X/\theta,\ x\in\overleftarrow{\delta}(b)\ \text{iff}\ y\in\overleftarrow{\delta}(b).$$

We remind the reader that $X/\theta\subset S(X)$ is the set of blocks of θ.

3.09 Lemma

For $\nu\in C(X)$, if $\nu\le\overleftarrow{\delta}(\nu)$, there exists a feedback $\phi:X\to U$ such that $\nu\le\nu\circ\delta_\phi$.

Proof. If $\nu\le\overleftarrow{\delta}(\nu)$, then for each block $b\in X/\nu$ we can choose a block $b'\in X/\nu$ such that $b\subset\overleftarrow{\delta}(b')$. Define ϕ such that $\delta_\phi(b)\subset b'$ for each $b\in X/\nu$. Then $x\equiv y\ [\nu]$ implies $\delta_\phi(x)\equiv\delta_\phi(y)\ [\nu]$, as required. Obviously, the ϕ so constructed is far from unique.

3.10 Proposition

Let $\nu^*:=(\theta\longmapsto\gamma\wedge\overleftarrow{\delta}(\theta))^{\downarrow}$. Then the DDP is solvable if $\overrightarrow{\varepsilon}(\nu^*)\le\nu^*$.

Proof. By virtue of being a fixpoint, $\nu^*=\gamma\wedge\overleftarrow{\delta}(\nu^*)$; hence

$$\nu^*\le\gamma \tag{3.101}$$

and

$$\nu^*\le\overleftarrow{\delta}(\nu^*) \tag{3.102}$$

By (3.102) and Lemma 3.09, there exists a feedback $\phi:X\to U$ such that

$$\nu^*\le\nu^*\circ\delta_\phi \tag{3.103}$$

Now define

$$\overrightarrow{\delta}_\phi:\theta\longmapsto\inf\{\sigma\in C(X)\,|\,x\equiv y\ [\theta]\ \text{implies}\ \delta_\phi(x)\equiv\delta_\phi(y)\ [\sigma]\}\,.$$

Then (3.103) implies

$$\vec{\delta}_{\phi}(\nu^*)\leq\nu^* \tag{3.104}$$

Now, for any $\theta\in C(X)$, $\vec{\varepsilon}_{\phi}(\theta) = \vec{\varepsilon}(\vec{\delta}_{\phi}(\theta))$.

$$\begin{aligned} \text{Thus} \qquad \vec{\varepsilon}_{\phi}(\nu^*) &= \vec{\varepsilon}(\vec{\delta}_{\phi}(\nu^*)) \\ &\leq \vec{\varepsilon}(\nu^*) && \text{(by 3.104)} \\ &\leq \nu^* \,. && \text{(by hypothesis)} \end{aligned}$$

By Lemma 3.07 and (3.101), it follows that ϕ solves the DDP.

3.11 Remarks

Proposition 3.10 provides a constructive criterion for solvability of the DDP that is, unfortunately, only sufficient. This is because, in general, the converse of Lemma 3.09 is not true. See [3] for more details.

4. DISCUSSION

The aim of feedback here is to achieve an appropriate invariance property of the system flow that is not present initially. This invariance amounts to a kind of stability for subsets, and decoupling for partitions (i.e. output maps).

5. REFERENCES

[1] W.M. Wonham. Linear Multivariable Control: A Geometric Approach, Lecture Notes in Economics and Mathematical Systems, vol. 101. New York: Springer-Verlag, 1974.

[2] W.M. Wonham. "Towards and abstract internal model principle", IEEE Trans. Systems, Man, and Cybernetics, vol. SMC-6, pp. 735-740, 1976.

[3] P. Liepa. Feedback Systems in a General Algebraic Setting. M.A.Sc. Thesis, University of Toronto, 1978. Also available as Systems Control Rpt. No. 7804, Dept. of Electrical Engineering, University of Toronto.

[4] S. MacLane and G. Birkhoff. Algebra. New York: MacMillan, 1967.

[5] B.S. Goh. "Nonvulnerability of ecosystems in unpredictable environments". Theoretical Population Biology, vol. 10, pp. 83-95, 1976.

CELLWORK TOPOLOGY, ITS NETWORK DUALS AND SOME APPLICATIONS THREE DIMENSIONAL KARNAUGH MAP AND ITS VIRTUAL PLANAR REPRESENTATION

H. L. LIU and K. S. FU
School of Electrical Engineering
Purdue University
West Lafayette, Indiana

ABSTRACT

This paper extends the definitions and properties of well known network topology to those for three dimensional case. Its network dual and retrieval dual, as defined later, are interesting and useful. The adjacency relationship of Karnaugh map can be effectively described through this approach.

I. INTRODUCTION

Network topology has been well developed and many useful results are being adopted by electrical engineers and operational researchers. This paper is aimed at extending the networks into higher dimensions. "Cellwork Topology," which is to study three dimensional topological figure from a network point of view, is proposed here.

The notation of this extension is exactly parallel to that of network topology. The definitions are stated in section 2. We find it particularly interesting to exploit the duals of a cellwork. The retrieval dual, which is the network dual of the cell to net dual, retrieves the information presented in te cellwork, and the floating retrieval dual is a retrieval dual which drops off some unimportant relationships aiming at simplifying the representation. These are presented in section 3.

The usefulness of this cellwork representation is yet to be discovered. The first application, the virtual planar representation of a three dimensional Karnaugh map, is presented in section 4. It is easily seen that two dimensional Karnaugh map can be explicitly represented by a network. It is expected that a three dimensional Karnaugh Map, which is composed of cubes, can be represented by a cellwork. The floating retrieval dual makes it easy and clear to draw a virtually planar graph, and once again we can work on minimization of the switching cost on this plannar map.

II. EXTENSION FROM NETWORK TO CELLWORK

Definitions:

*[D.1] Planate Graph: (in contrast to Linear Graph)

A "planate graph" is a set of (curved) planes called "Walls," and lines called "edges" which are the end-edges of some walls. The point intersection of edges is called "corner."

*[D.2] The degree of an edge is the number of walls incident with that edge.

*[D.3] A graph G' containing a subset of the elements of a given gragh G is called a subgraph of G.

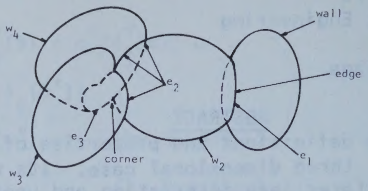
*[D.4] Hall: (in contrast to Path)

If a set of the elements in a graph constitute a simple (curved) plane, this set is called a "hall." The end-edges of this set are called "windows."

*[D.5] Closure (in contrast to Circuit)

If a "hall" has an enclosed volume and we cannot find any closed path (circuit) on this hall that does not divide the hall into two, it is called a "closure." A set of all the "closures" of a graph is called a "closure set."

Example 1:

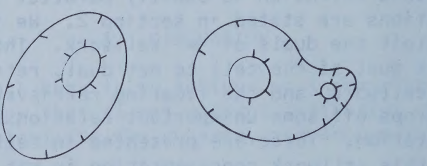


The whole Graph is a closure. w_1, w_2, e_1, e_2 constitute a hall and e_2 is the window of the hall.

*[D.6] If a hall has an enclosed volume and we can find a circuit disjoint from the hall and while shrinking this circuit, it has to touch the hall, then this hall is called a "Torus."

Example 2:

Tori:



*[D.7] A graph is "connected" if we can get to any point on it without leaving the graph.

*[D.8] Two graphs are disjoint if their edges do not intersect.

Note:(Seemingly touched graphs might be disjoint, because they are not touched in higher dimensional graph, if their edges do not intersect. This class of "seemingly touched graph" is ruled out in the foregoing discussion, since it is of no interest.)

*[D.9] Body (in contrast to closed area)
Any separated volume by a graph is called a "body."

*[D.10] A graph is "wall-oriented" if on each wall there is an orientation in the form of an arrow from one body to another.

A graph is "edge-oriented" if on each edge there is an orientation in the form of an arrow from one wall to another.

A graph is "corner oriented" if the corners and edges of a graph constitute an oriented linear graph. (Graphs whose edges do not form a legitimate network is not discussed in this argument).

*[D.11] A "Tree" of a connected graph G is a maximal connected subgraph that contains all the edges of G but no closure. The walls contained in a tree is called branches of that tree.

*[D.12] A cut set of a connected graph is a minimal set of wall(s) which together with their attached edges separating the graph into two.

*[D.13] Congruence: Two graphs are congruent if there is a 1 to 1

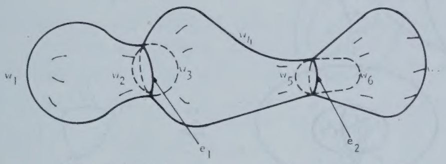
correspondence between their walls and edges, such that the incident relationship are preserved.

Properties:

Δ[1] There exists at least one tree in every connected graph.

Δ[2] Let A be the Incident matrix of G. B be the Closure matrix of G. then $A B^t = 0$. (Where A. B is defined as visualized from the following example:)

Example 3:



Incident Matrix A:

	w_1	w_2	w_3	w_4	w_5	w_6	w_7
e_1	1	1	1	1	0	0	0
e_2	0	0	0	1	1	1	1

Closure Matrix B:

	w_1	w_2	w_3	w_4	w_5	w_6	w_7
Cl_1	1	0	1	0	0	0	0
Cl_2	0	1	0	1	0	1	0
Cl_3	0	0	0	0	1	0	1
--	-	-	-	-	-	-	-

·

redundant

·

$Cl_1 = w_1 \cup w_3 \cup e_1$

$Cl_2 = w_2 \cup w_4 \cup w_6 \cup e_1 \cup e_2$

$Cl_3 = w_5 \cup w_7 \cup e_2$

* proof of $AB^t = 0$:

$AB^t = [P_{ij}]$ or $P_{ij} = \sum_{K=1}^{W} a_{iK}b_{Kj}{}^{T}$ where b^T_{Kj} is the Kj element of B^T. W is the number of walls. Number of walls common to any edge and any closure is either 0 or 2. (i.e. 0 if e is not thoroughly on B_j, 2 if on B_j). Hence, $P_{ij} = 0$, or 2(mode 2) = 0.

Δ Many other relations can be suggested and organized.

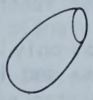
III. DUALS

Definitions:

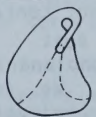
* [D.14] Body-like (in contrast to planar).

If every wall of a planate graph is thoroughly attached to at least two bodies, it is "body-like."

Example 4: The "bag" and "Klein bottle" are not body-like:



bag



Klein bottle

*[D.15] Cell to Net Dual: (c to n dual)

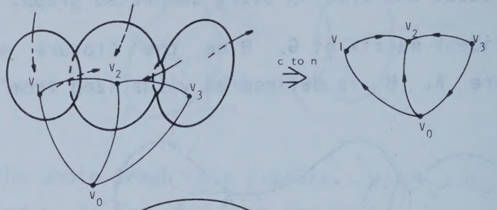
For a planate graph that is body-like we assign a vertex to every body. Then for each pair of bodies that have a commonly attached wall, we draw a branch connecting the corresponding vertices. The orientation (if any) is the same as the wall. The corresponding linear graph is called a "Cell to Net Dual."

*Net to Cell Dual is defined the reversed way.

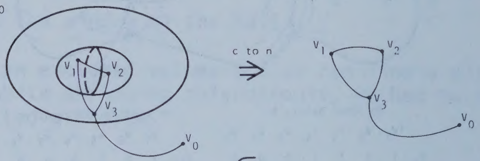
*[D.16] Planar Planate

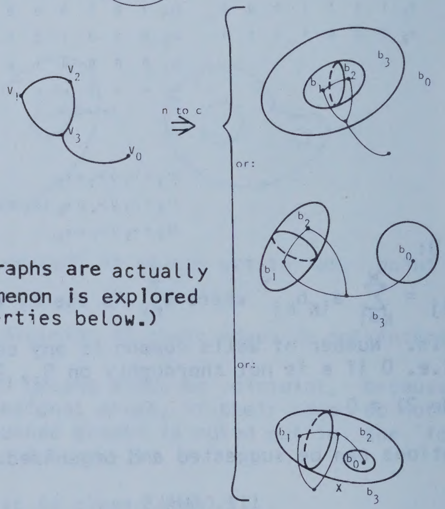
If the c to n dual is a planar linear graph, we call the corresponding planate graph a "planar planate graph."

Example 5:



Example 6:





These three planate graphs are actually congruent. This phenomenon is explored in the first two properties below.)

*[D.17] Retrieval Dual

The linear dual of the cell to net dual of a planate graph is called a retrieval dual.

(Here linear dual is defined in a broader sense i.e., we draw a circuit through a branch which is not contained in any circuit.)

*[D.18] Floating Retrieval Dual

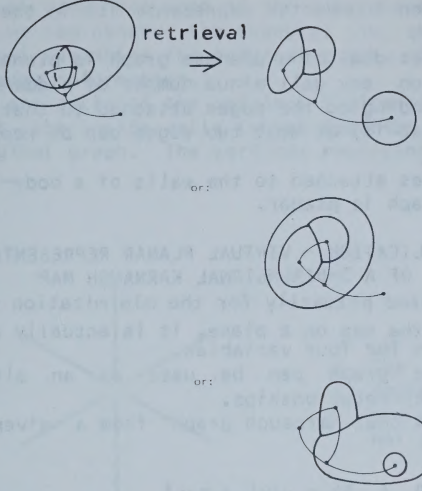
In many applications, nonplanarity arises from only one vertex (representing one body). We omit this vertex and form the linear dual, which is called "floating retrieval dual."

Properties:

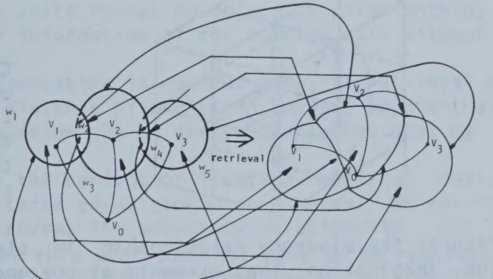
Δ[1] Any body-like graph can be projected onto a 4-dimensional sphere and preserve its body-like nature. A graph on the 4-dimensional sphere can also be projected onto a 3-dimensional universe, and preserve its nature of body-like.

Δ[2] Any body-like graph can be "twisted" in such a 4-dimensional way that any one of its bodies becomes outside universe (which is also a body) and

Example 7:



Example 8:



at the same time the original outside universe becomes inside. The graph is congruent with the original one.

Δ[3] ~ Δ[7]: Relations of Correspondence

Δ[3] An enclosed region of the cell to net dual represents some edge of the planate graph.

Δ[4] The regions enclosed in the branches together with the outside region of the retrieval dual has a one to one correspondence with the bodies of the original planar planate graph.

Δ[5] The branches of the retrieval dual have a one to one correspondence with the walls of the original planar planate graph.

Δ[6] The vertices in the retrieval dual correspond to some edges in the original planar planate graph. And number of vertices in the retrieval dual is less or equal to that of edges.

Δ[7] If three walls have a common edge, and the branches corresponding to these walls in a cell to net dual do not form a circuit, (and then the corresponding branches in the retrieval dual do not intersect to a vertex) then the edge is hidden (no vertex represents it) in the retrieval dual.

Δ[8] If the cell to net dual of a planate graph is planar, then its number of edges attached on any wall minus number of hidden vertice on the retrieval dual corresponding to the edges attached to that wall is less than or equal to two. (That is, at most two edges can be represented).

Δ[9] If number of edges attached to the walls of a body-like graph is at most two, then the graph is planar.

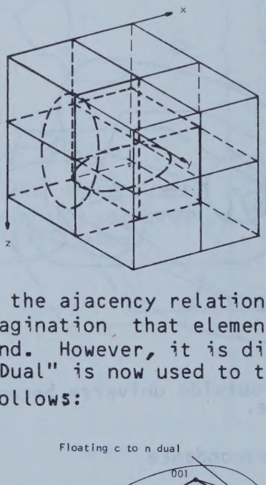
IV. APPLICATION - VIRTUAL PLANAR REPRESENTATION OF A 3-DIMENSIONAL KARNAUGH MAP

Karnaugh map is used primarily for the minimization of switching function cost. While we draw the map on a plane, it is actually a cylinder for three variables, and a torus for four variables.

Now, planar planate graph can be used as an alternative tool to represent the adjacency relationships.

We form a "3-dimensional Karnaugh graph" from a given truth table as below:

Example 9: $f(z\ y\ x) = \sum(0, 4, 5) = y'x' + z y'$

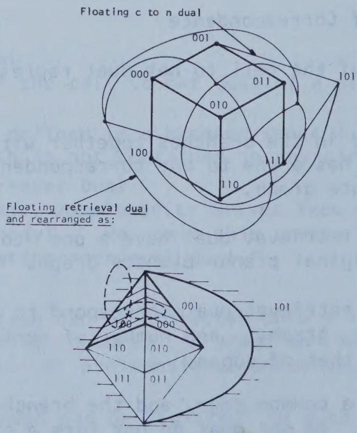


Cubes with "1" are outlined in thick lines. Two "covers" are drawn in thick broken circles.

In the above figure, the ajacency relationship is clearly represented without using the imagination that elements at one end are glued to the elements at the other end. However, it is difficult to draw.

"Floating Retrieval Dual" is now used to transform this graph onto a two dimensional plane, as follows:

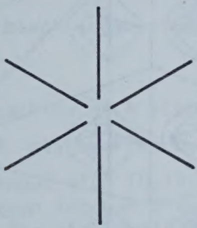
Example 10:



Now, all the adjacency relationships are preserved in this retrieval dual. Actually this dual describes most of the topolgical information of the original graph. (As we can observe the examples in section three, nonfloating retrieval dual resembles the original graph. It looks skewed, because it drops some edge information.)

The divided areas are cells of the original 3 dimensional Karnaugh map. The lines represent the inside walls (which describe the adjacency relationship) of the original graph. The vertices represent part of the edges of the walls.

Example 11:



The six vertices represent the axes in the middle of the original graph.

Since the outside walls reveal no adjacency relationship, the floating retrieval drops the information of the outside wall without losing any adjacency relationship.

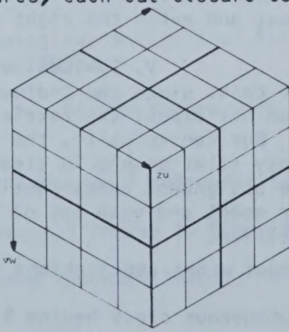
The logical representation is now put into the retrieval dual. We can observe that the distance of digits in any two adjacent areas is exactly one. the logical equation can be derived from this dual by "covering the "true" areas".

In conclusion, by the concept of floating retrieval dual, we reach a planar graph with three edges around each areas. We can now work on this dual, which clearly reveal the adjacency relationship. The "state assignment" of each area of this dual does not have to be restricted to the one presented in the example, since the graph can be twisted 3-dimensionally to get all different congruent graphs.

We further investigate the 3 dimensional Karnaugh map with six variables. Since we need to draw a graph which represents 6 adjacency relationships for every element, the cell-work of the piling of cubes seems useful. We pile up 64 cubes into one big cube , then by imaging the surfaces of this big cube touching end to end, we observe each cube has six neighboring cubes.

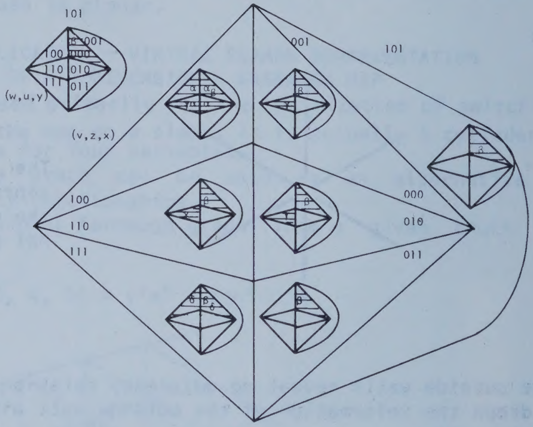
To apply the result obtained in the previous discussion, we first cut this big cube into eight closures, each cut closure contains eight cubes.

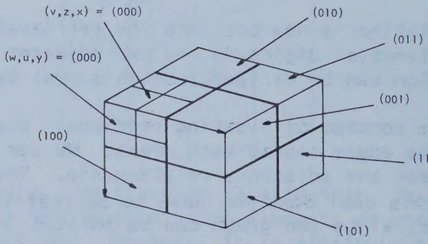
Example 12: Subdividing the graph



We treat this graph as eight closures and draw its floating retrieval dual. This dual is the same as that obtained in the previous discussion. By this treatment, we dropped some adjacency relationships: (1) The outmost closure now contains adjacency relationship of the outmost cubes, so the "floating" dropped these relationships, (2) By terating 4 walls of 4 adjacent cubes as one wall, we could not specify the individual correspondence. These dropped relationships can be put back to the retrieval map by right positioning, as described below.

Example 13:





f (v,z,x,w,u,y) = 2(0,6,8,16,22,24,32,34,36,38,40,48,54, 56,60) = vz'x'y' + w'u'y' + x'wuy' + vzxu'y'

For each closure, we draw a floating retrieval dual corresponding to the eight cubes it contains. Now three of the adjacent elements for each element can be observed on its "local" retrieval dual. and we can find the remaining three adjacent elements by going into the adjacent area of the "global" retrieval dual and match the right position (same shape as plotted).

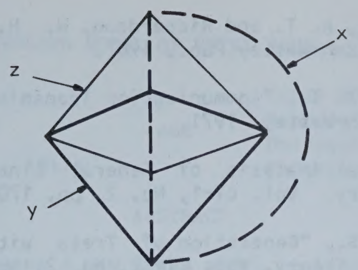
V. CONCLUSION

For three variable case, since the 2-dimensional Karnaugh map is indeed very handy to use, we can not expect the retrieval dual of 3-dimensional graph to be easier to use. But conceptually, the shaped map is very helpful. The adjacency relationship is clearly revealed on the edges. For the concept of "state assignment using partitions," we can draw sets of closed curves on our model and each set of closed region represent (and is indeed itself) a partition.

Some interludes about other applications of this map are stated on the following page:

1) To represent an autonomous clock having 8 states. The partitions ζ_X,

Example 14:



ζ_Y, ζ_Z are all of 2 blocks each of 4 states. We assign X, Y, Z to ζ_X, ζ_Y, ζ_Z respectively.

2) To represent a Mealy machine of 4 states. The partitions ζ_Y, ζ_Z are of 2 blocks each of 2 states. We assign Y Z to ζ_Y ζ_Z respectively. X represent the single input with {0,1}. We draw a closed curve and divide the 4 states graph into an 8-region map.

So we can now work on the retrieval dual with regions visualized as states or partitions while keeping full track of all the adjacency relationships.

For six-variable case, a 2-dimensional map has to put eight combinational representation on each side of the map and the adjacency relationship becomes very confusing and unclear. A nested map of the retrieval dual of the 64-cubic - 3- dimensional Karnaugh map now proves to be much easier and clearer. er. by crossing the edges and matching the position, we can easily visualize the 6 adjacent elements for each element. The minimal function of $f(y,u,w,x,z,v) = \sum(0,6,8,16,22,24,32,34,36,38,40,48,54,56,60)$ was easily worked out to be vz' x'y' + w'u'y' + x'wuy' + vz u'y', as shown in the figure.

There is no confusion and difficulty to draw our model at all. After that sketch, we can clearly assign states or variables and functions to the graph which reveal more information than the 2-dimensional Karnaugh map does, and we can more systematically set up the covers and find the minimal function cost.

All these operations comes from the idea of the "retrieval dual" of a 3-dimensional Karnaugh map. The creation of "floating retrieval dual" and the investigation of its properties are entirely based on the developed "Cellwork Topology," as described in section II and section III.

REFERENCES

1. Chan, S. P., "Introductory Topological Analysis of Electrical Networks," Holt, Rinehart and Winston Inc., 1969.

2. Johnson, D. E. and Johnson, J. R., "Graph Theory with Engineering Applications," Ronald Press, 1972.

3. Brown, R., "Elements of Modern Topology," McGraw-Hill, 1968.

4. Barr, S., "Experiments in Topology," Thomas Y. Crowell Company, 1964.

5. Kohavi, Z., "Switching and Finite Automata Theory," McGraw-Hill, 1970.

6. Shearer, J. L., Murhhy, A. T. and Richardson, H. H., "Introduction to System Dynamics," Addison-Wesley Pub., 1967.

7. Frank, H. and Frisch, I. T., "Communication Transmission and Transportation Networks," Addison-Wesley, 1971.

8. Talbot, A., "Topological Analysis of General Linear Networks," IEEE Trans. on Circuit Theory. Vol. CT-1, No. 2, pp. 170-180, 1965.

9. Meyeda, W. and Seshu, S., "Generation of Trees without Duplications," IEEE Trans. on Circuit Theory, Vol. CT-12, No. 2, pp. 181-185, 1965.

10. Dunn, W. R., Jr. and Chan, S. P., "An Algorithm for Testing the Planarity of a Graph," IEEE Trans. on Circuit Theory, Vol. CT-15, No. 2, pp. 166-168, 1968.

PT-MARKED GRAPHS: A REDUCED MODEL OF PETRI NETS

STONE H. YU
Honeywell Inc.
4849 North Scott
Schiller Park, IL 60176

and

TADAO MURATA
Dept. of Information Eng.
University of Illinois
Chicago, IL 60680

ABSTRACT

This paper introduces a new graph model, called PT-marked graphs, which is essentially marked graphs with an additional type of vertices. These new graphs can be used as a reduced representation for Petri nets since the two graph models are equivalent with respect to many properties.

1. INTRODUCTION

Petri nets [2,5,8,10,13] and related graph models have been used to represent a wide variety of systems. A survey of some of their applications can be found in [1]. However, just as the systems they represent, Petri nets sometimes become too large to be analyzed efficiently. In such cases it is desirable to simplify Petri nets without altering any relevant properties. Prior efforts toward this goal include macro E-nets used by Noe and Nutt [12], Petri net reductions as considered by Kinney and Han [6], and Pro-nets described by Noe [11].

This paper introduces a new approach and a new model, PT-marked graphs, which is essentially marked graphs [9,10] with a new type of vertices. PT-marked graphs are directed graphs with zero or more tokens assigned to each arc, and have two types of vertices: P- and T-vertices. These vertices are respectively depicted as circles and squares in figures. A P-vertex (T-vertex) is firable when there is at least a token on any (each) one of its incoming arcs. The firing of a P-vertex (T-vertex) removes a token from any (each) one of its incoming arcs, and adds a token to any (each) one of its outgoing arcs. T-vertices without incoming arcs are always firable, but P-vertices lacking either incoming or outgoing arcs are defined to be nonfirable.

Vertices of either type which have single incoming and single outgoing arcs are called elementary. Section 3 demonstrates that elementary vertices may be added or removed without affecting many PT-marked graph properties. Section 4 introduces a methodology for representing Petri nets by PT-marked graphs, and shows that many Petri net properties are preserved by this transformation. These results establish a new technique of simplifying Petri nets while preserving many of their properties. PT-marked graphs are also a suitable model for many systems, and Section 5 exhibits a situation where PT-marked graphs are a more adequate representation than Petri nets.

2. BASIC TERMINOLOGY

Many PT-marked graph terminologies are adopted from Petri nets, where places correspond to arcs and transitions to vertices. A series of legal vertex firings is called a firing sequence. The assignment of zero or more tokens to each arc constitutes a marking. A marking M_n is reachable from a marking M_0 if there exists a firing sequence transforming M_0 to M_n. When there is an arc directed from vertex v_i to vertex v_j, v_i is called a predecessor of v_j and v_j a successor of v_i.

This material is based upon work supported by NSF Grant ENG 78-05933.

A PT-marked graph G at marking M_0, denoted as PT-MG(G,M_0), is n-bounded if each arc of G has at most n tokens for every marking reachable from M_0. Specifically, PT-MG(G,M_0) is safe if it is 1-bounded. Four types of liveness are distinguished as in [2]. A vertex v in a PT-MG(G,M_0) is L1 if it appears in some firing sequence starting at M_0. v is L2 if, for every positive integer n, it appears at least n times in some firing sequence starting at M_0. v is L3 if it appears infinitely often in some infinite firing sequence starting at M_0. v is L4 if it is L1 for all markings reachable from M_0.

A PT-MG(G,M_0) is consistent if there is some firing sequence starting at M_0 which fires each vertex in G at least once and whose final marking is also M_0. The weighted token number of an arc is the number of tokens on the arc times a preassigned weight of that arc, and the sum of all weighted token numbers is called the weighted token sum. A PT-MG(G,M_0) is invariant if, for some set of arc weights, its weighted token sum remains constant for all markings reachable from M_0.

3. A HIERARCHICAL REPRESENTATION

Consider a PT-MG(G,M_0) with an elementary vertex v_e. Construct a PT-MG(H,M_0') by removing v_e while joining its incoming arc a_i and its outgoing arc a_o. Also lump together the tokens of a_i and a_o on the new arc. This PT-MG(H,M_0') is then called an abbreviation of PT-MG(G,M_0). To illustrate, the PT-marked graph in Fig. 2 is an abbreviation of the one in Fig. 1 with respect to elementary vertices t_3 and p_3. To facilitate discussion, let $v_s(v_p)$ be the successor (predecessor) of v_e in G, and $t_s(t_p)$ be its corresponding vertex in H. The f-firing of t_s, denoted as $f(t_s)$, is either v_s or the firing sequence $v_e v_s$. The f-firing of any other vertex t in H, denoted as f(t), is its corresponding vertex in G.

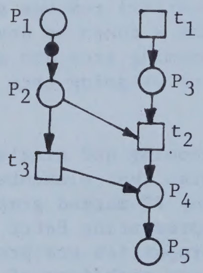


Fig. 1. A PT-marked graph example

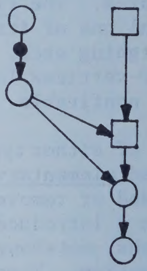


Fig. 2. An abbreviation of the PT-marked graph in Fig. 1

Lemma 1: A vertex t in PT-MG(H,M_0') is firable if and only if f(t) in PT-MG(G,M_0) is firable. The marking after firing t is an abbreviation of the marking after firing f(t).

Proof: Firing t_s in H corresponds to firing either v_s or $v_e v_s$ in G, depending on whether a_o is marked at M_0.

The rest follows directly from the constructed relationship between H and G.

Lemma 2: A series of vertices t_1, t_2, ..., t_m is a firing sequence for PT-MG(H,M_0') if and only if $f(t_1)$, $f(t_2)$, ..., $f(t_m)$ is a firing sequence for PT-MG(G,M_0). The final marking for H is an abbreviation of the final marking for G.

Proof: This follows from Lemma 1 by induction.

Lemma 3: Every firing sequence of PT-MG(G,M_0) can be rearranged into form $s_f s_{v_e}$, where s_f is a series of f-firings and s_{v_e} is a series of v_e firings. Moreover, this rearrangement will not affect the final markings.

Proof: Let $S = \{s|s$ is a firing sequence of G$\}$, and $F = \{s \in S|s$ can be rearranged into form $s_f s_{v_e}\}$. $F \subseteq S$ by definition. Assume $F \subset S$ and let x be the shortest member in $S - F$. Suppose $x = yv$ where v is the last vertex in x. $y \in F$; otherwise, $y \in S - F$ and is shorter than x, a contradiction. So y can be rearranged as $s_f s_{v_e}$. v cannot be v_e, else x can be rearranged as $s_f s_{v_e}$, a contradiction. If v is v_s, v_s is already firable either before or after the first v_e in s_{v_e}. Then x can be rearranged as $s_f f(t_s) s_{v_e}$; a contradiction. If v is any other vertex, it is already firable before s_{v_e}. Then x can be rearranged as $s_f f(t) s_{v_e}$ where $v = f(t)$, a contradiction. Therefore, $F \not\subset S$ and $S = F$.

Any firing sequence and its rearrangement have identical final markings since precisely the same set of vertices are involved.

Theorem 1: A vertex t in PT-MG(H,M_0') is L2 if and only if f(t) in PT-MG(G,M_0) is L2.

Proof: Assume t is L2. Then, for any n, there is a firing sequence starting at M_0' which contains t at least n times. By Lemma 2, for any n, there is a firing sequence starting at M_0 which contains f(t) at least n times. Hence f(t) is L2.

The converse follows by applying above arguments in the reverse order.

Corollary 1: A vertex t in PT-MG(H,M_0') is L1 if and only if f(t) in PT-MG(G,M_0) is L1.

Proof: This follows from Theorem 1 by setting n to 1.

Corollary 2: A vertex t in PT-MG(H,M_0') is L3 if and only if f(t) in PT-MG(G,M_0) is L3.

Proof: This follows from Theorem 1 by letting n go to infinity.

Theorem 2: A vertex t in PT-MG(H,M_0') is L4 if and only if f(t) in PT-MG(G,M_0) is L4.

Proof: Assume f(t) is L4 but t is not L4. Then t is not L1 for some marking M_1' reachable from M_0'. By Lemma 2 there is a marking M_1 reachable from M_0, where M_1' is an abbreviation of M_1. f(t) is L4 at M_0, so f(t) is L1 at M_1. Then, by Corollary 1 t is L1 at M_1'; a contradiction. Hence, t is L4 if f(t) is L4.

The converse follows similarly, together with the use of Lemma 3.

Theorem 3: PT-MG(G,M_0) is n-bounded if and only if PT-MG(H,M_0') is n-bounded.

Proof: Assume M_0' is not n-bounded. Then there is a firing sequence starting at M_0' whose final marking M_1' has more than n tokens on some arc. By Lemma 2 there is a firing sequence starting at M_0 leading to a marking M_1 where M_1' is an abbreviation of M_1. At marking M_1 either there is an arc in G containing more than n tokens, or the sum of tokens on a_i and a_o is greater than n. In the latter case v_e can fire until a_o has more than n tokens. Hence M_0 is not n-bounded.

Conversely, assume M_0 is not n-bounded. Then there is a firing sequence s starting at M_0 whose final marking M_1 has more than n tokens on some arc. By Lemma 3 s can be rearranged into form $s_f s_{v_e}$. Let M_2 be the final marking of s_f. From Lemma 2 there is a firing sequence starting at M_0' whose final marking M_2' is an abbreviation of M_2. M_2' is also an abbreviation of M_1. So some arc in H has more than n tokens at M_2'. Hence M_0' is not n-bounded.

Corollary 3: PT-MG(G,M_0) is safe if and only if PT-MG(H,M_0') is safe.

Proof: This follows from Theorem 3 by setting n to 1.

Theorem 4: PT-MG(G,M_0) is consistent if and only if PT-MG(H,M_0') is consistent.

Proof: Assume PT-MG(G,M_0) is consistent. Then there is a firing sequence s starting at M_0 which fires every vertex in G and whose final marking is M_0. By Lemma 3 s has form $s_f s_{v_e}$. Let M_1 be the final marking of s_f. By Lemma 2 there is a firing sequence starting at M_0' which fires every vertex in H and whose final marking M_1' is an abbreviation of M_1. M_1 and M_0 have identical abbreviations, so $M_1' = M_0'$. Hence PT-MG(H,M_0') is consistent.

Conversely, assume PT-MG(H,M_0') is consistent. Then there is a firing sequence s' starting at M_0' which fires every vertex in H and whose final marking is M_0'. By Lemma 2 there is a firing sequence s starting at M_0 which fires every f-firing in G and whose final marking is, say, M_1. M_0' is an abbreviation of both M_0 and M_1. So every arc, except a_i and a_o has identical markings at M_0 and M_1. t_p and t_s must fire equally often in s'; else initial and final markings of s' cannot be identical. So v_p and v_s occur equally often in s. Since the abbreviation of a marking is not changed by v_e firings, it can be assumed that s contains v_p, v_e, and v_s the same number of times. Hence a_i and a_o have identical markings at M_0 and M_1. Thus $M_0 = M_1$ and PT-MG(G,M_0) is consistent.

Theorem 5: PT-MG(G,M_0) is invariant if and only if PT-MG(H,M_0') is invariant.

Proof: Assume PT-MG(H,M_0') is invariant. Assign a_i and a_o with the weight of the arc from t_p to t_s in H. Assign all other arcs in G with weights of their corresponding arcs in H. Firing v_e does not change weighted token sum since a_i and a_o have the same weight. Firing any other vertex in G does not change weighted token sum, because firing its corresponding vertex in H does not change weighted token sum. Hence PT-MG(G,M_0) is invariant.

Conversely, assume PT-MG(G,M_0) is invariant. a_i and a_o must have equal weights; otherwise G cannot be invariant. Assign the arc from t_p to t_s in H with the weight of a_i. Assign all other arcs in H with weights of their corresponding arcs in G. By the same arguments as in the previous part of this proof, PT-MG(H,M_0') is invariant.

As a consequence of the results in this section, the removal of elementary vertices does not alter the foregoing properties. If G_1 is constructed by adding an elementary vertex to PT-marked graph G_2, then G_2 is an abbreviation of G_1. It follows that the addition of elementary vertices also does not alter the aforementioned properties. Therefore, adding or removing elementary vertices allows PT-marked graphs to hierarchically represent systems at any desired level of detail while preserving boundedness, safeness, L1, L2, L3, L4, consistence, and invariance properties.

4. REPLICA PT-MARKED GRAPHS

The replica of a Petri net marking is a PT-marked graph marking constructed by first moving tokens from places to their incoming arcs, and then substituting a P-vertex for each place and a T-vertex for each transition. A Petri net place without incoming (outgoing) arcs is called a source (sink) place. For any source (sink) place, a predecessor (successor) P-vertex is added to the corresponding P-vertex in the replica. For instance, the PT-marked graph in Fig. 1 is a replica of the Petri net in Fig. 3. Initial tokens in a source place represent control or data signals of external origin which is not modeled by the Petri net. So it is reasonable to introduce a corresponding predecessor P-vertex in the replica, as its outgoing arc will not acquire additional tokens after the initial marking. Similarly, it is reasonable to introduce a corresponding successor P-vertex for each sink place.

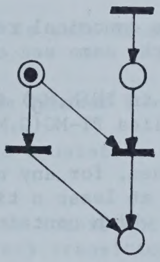


Fig. 3. A Petri net corresponding to the PT-marked graph in Fig. 1.

The canonical firing unit in a replica PT-marked graph is a firing sequence which fires a T-vertex t after all its predecessor P-vertices have fired exactly once toward t. (t_1), (p_2,p_3,t_2) and (p_2,t_3) in Fig. 1 are examples of canonical firing units. Hereafter, PN(N,M) denotes a Petri net N at marking M.

Lemma 4: Let PT-MG(G,M') be a replica of PN(N,M).

(1) A transition t in N is firable at M if and only if the canonical firing unit T of its corresponding T-vertex t' in G is firable at M'.

(2) The marking after firing T is a replica of the marking after firing t.

Proof: Assume t is firable. Then each input place of t has tokens. So each predecessor P-vertex of t' has tokens on its incoming arcs and is firable. Hence T is firable.

Conversely, assume T is firable. Then each predecessor P-vertex of t' is firable and has tokens on its incoming arcs. Hence each input place of t has tokens, and t is firable.

(2) follows directly from the construction of Petri net replicas.

Lemma 5: Let PT-MG(G,M') be a replica of PN(N,M). A series of transitions $t_1, t_2, \ldots, t_m$ is a firing sequence for PN(N,M) if and only if $T_1, T_2, \ldots, T_m$ is a firing sequence for PT-MG(G,M'), where T_i is the canonical firing unit corresponding to t_i. Furthermore, the final marking for G is a replica of the final marking for N.

Proof: This follows directly from Lemma 4 by induction.

Lemma 6: Every firing sequence of a replica PT-MG(G,M_0') has a canonical rearrangement in the form $s_c s_p$ where s_c is a series of canonical firing units and s_p is a series of P-vertices. Moreover, this rearrangement will not affect the final markings.

Proof: Let $S = \{s | s \text{ is a firing sequence of } G\}$, and $K = \{s \varepsilon S | s \text{ has a canonical rearrangement in the form } s_c s_p\}$. $K \subseteq S$ by definition. Assume $K \subset S$ and let x be the shortest member in $S - K$. Suppose $x = yv$ where v is the last vertex in x. $Y \varepsilon K$; otherwise, $y \varepsilon S - K$ and is shorter than x, a contradiction. So y can be rearranged into form $s_c s_p$. v is not a P-vertex; otherwise $s_c s_p v$ is a canonical rearrangement of x, a contradiction. Thus v is a T-vertex. A P-vertex firing does not enable or disable any other P-vertex; so s_p can be rearranged such that all the predecessor P-vertices of v are at the beginning. v is firable as soon as all its predecessor P-vertices have fired; so $s_p v$ can be rearranged into canonical form. Then $s_c s_p v$ or x has a canonical rearrangement, a contradiction. Therefore, $K \not\subset S$ and $S = K$.

Any firing sequence and its canonical rearrangement have identical final markings since precisely the same set of vertices are involved.

Theorem 6: A transition t in PN(N,M_0) is L2 if and only if its corresponding T-vertex t' in the replica PT-MG(G,M_0') is L2.

Proof: Assume t is L2. Then, for any n, there is a firing sequence starting at M_0 which contains t at least n times. By Lemma 5 there is a firing sequence starting at M_0' which contains t' at least n times. So t' is L2.

Conversely, assume t' is L2. Then, for any n, there is a firing sequence s' starting at M_0' which contains t' at least n times. By Lemma 6 s' can be rearranged into form $s_c s_p$ where s_c contains t' at least n times. So, by Lemma 5 there is a firing sequence starting at M_0 which contains t at least n times. Hence t is L2.

Corollary 4: A transition t in PN(N,M_0) is L1 if and only if its corresponding T-vertex t' in the replica PT-MG(G,M_0') is L1.

Proof: This follows from Theorem 6 by setting n to 1.

Corollary 5: A transition t in PN(N,M_0) is L3 if and only if its corresponding T-vertex t' in the replica PT-MG(G,M_0') is L3.

Proof: This follows from Theorem 6 by letting n go to infinity.

Theorem 7: A transition t in PN(N,M_0) is L4 if its corresponding T-vertex t' in the replica PT-MG(G,M_0') is L4.

Proof: Suppose, on the contrary, t is not L4. Then t is not L1 for some marking M_1 reachable from M_0. By Lemma 5 there is a marking M_1'

reachable from M_0', where M_1' is a repliac of M_1. t' is L4 at M_0', so t' is L1 at M_1'. Then, by Corollary 4 t is L1 at M_1; a contradiction. Hence t is L4.

The converse is not necessarily true; a counterexample will be presented in Section 5.

Theorem 8: A PN(N,M_0) is n-bounded if and only if its replica PT-MG(G,M_0') is n-bounded.

Proof: Assume M_0 is not n-bounded. Then there is a firing sequence starting at M_0 whose final marking M_1 has more than n tokens in some place p. By Lemma 5 there is a firing sequence starting at M_0' leading to a replica marking M_1'. Then the P-vertex p' corresponding to p has more than n tokens on its incoming arcs at M_1'. p' can move n+1 tokens to one of its outgoing arcs. Hence M_0' is not n-bounded.

Conversely, assume M_0' is not n-bounded. By Lemma 6 there is a firing sequence $s_c s_p$ starting at M_0' which moves more than n tokens to some arc c. If c is an incoming arc of some P-vertex, then c already has more than n tokens before s_p fires. If c is an outgoing arc of some P-vertex p, then p has more than n tokens on its incoming arcs before s_p fires. In either case, the final marking of s_c has more than n tokens on the incoming arcs of some P-vertex p_1. By Lemma 5 there is a firing sequence starting at M_0 which moves more than n tokens to the place corresponding to p_1. So M_0 is not n-bounded.

Corollary 6: A PN(N,M_0) is safe if and only if its replica PT-MG(G,M_0') is safe.

Proof: This follows from Theorem 8 by setting n to 1.

Theorem 9: A PN(N,M_0) is consistent if and only if some of its replica PT-MG(G,M_0') is consistent.

Proof: Assume PN(N,M_0) is consistent. Then there is a firing sequence starting at M_0 which fires every transition in N and whose final marking is M_0. By Lemma 5 there is a firing sequence s' starting at M_0' which fires every canonical firing unit, hence every vertex, in G. The final marking M_1' of s' is a replica of M_0. Since M_0' is also a replica of M_0, M_0' can be constructed such that it is identical to M_1'. So PT-MG(G,M_0') is consistent.

Conversely, assume PT-MG(G,M_0') is consistent. Then there is a firing sequence s' starting at M_0' which fires every vertex in G and whose final marking is M_0'. By Lemma 6 s' has a canonical rearrangement s_1'. Any T-vertex and all of its predecessor P-vertices must fire equally often in s'; otherwise s' cannot have identical initial and final markings. Thus s_1' is a series of canonical firing units which fires every canonical firing unit in G. By Lemma 5 there is a firing sequence s which fires every transition in N and whose initial and final markings are both M_0. So PN(N,M_0) is consistent.

Theorem 10: A PN(N,M_0) is invariant if and only if its replica PT-MG(G,M_0') is invariant.

Proof: Assume N is invariant. For each P-vertex in G, assign to its incoming and outgoing arcs the weight of its corresponding place in N. Firing any P-vertex in G does not change the weighted token sum since its incoming and outgoing arcs have equal weights. Firing a T-vertex in G is identical to firing its corresponding transition in N except for token locations. Transition firings do not change the weighted token sum of N by assumption, so T-vertex firings do not change the weighted token sum of G. Hence G is invariant.

Conversely, assume G is invariant. A P-vertex can move a token from any one of its incoming arcs to any one of its outgoing arcs. So all incoming and outgoing arcs of a P-vertex must have the same weight, otherwise the weighted token sum could change. For each place in N, assign to it the weight of the incoming arcs of its corresponding P-vertex in G. Then, the invariance of N follows from the same arguments as in the previous part of this proof.

As a consequence of the theorems stated in this section, Petri net analyses are reducible to PT-marked graph analyses except for L4. The converse is also true, since elementary vertices can be added to a PT-marked graph until a Petri net replica is constructed. Therefore, the two graph models are equivalent with respect to boundedness, safeness, L1, L2, L3, consistence, and invariance.

5. PT-MARKED GRAPH APPLICATIONS

P- and T-vertices describe logical disjunction and conjunction, so PT-marked graphs are well suited to represent many systems. Flowcharts, PERT charts [3], and syntactic diagrams [14] are just a few examples of commonly used models which are PT-marked graphs with only notational differences.

Producer-consumer systems together with their semaphores are representable by Petri nets [7], and hence by replica PT-marked graphs. A semaphore s, as defined in [4], is a nonnegative integer accessible by two operations: P(s): wait until $s > 0$, then set $s := s - 1$; and v(s): $s := s + 1$.

Consider the producer-consumer system S shown in Fig. 4, its Petri net representation N in Fig. 5, and its PT-marked graph representation G in Fig. 6. Miller [7] noted that the system S has a deadlock caused by P2 performing a P(s1) and P3 performing a P(s2). The PT-marked graph model G has a corresponding reachable deadlock as depicted in Fig. 7. The Petri net model N, on the other hand, has no deadlocks. So this is a counterexample for the converse of Theorem 7. More significantly, this example demonstrates that producer-consumer systems, while not adequately representable by Petri nets, can be accurately modeled by PT-marked graphs.

```
P1: P(s3)     P2: P(s1)     P3: P(s2)
     ...          P(s2)         P(s1)
    V(s1)          ...           ...
    V(s2)         V(s3)         V(s3)
```

Fig. 4. A producer-consumer system

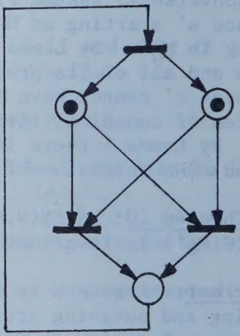


Fig. 5. A Petri net model of the system in Fig. 4

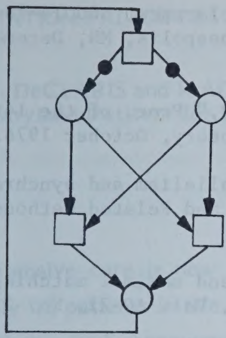


Fig. 6. A PT-marked graph model of the system in Fig. 4

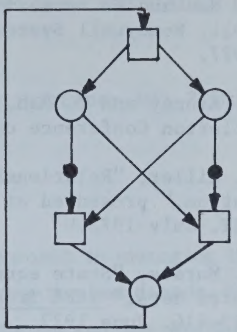


Fig. 7. A PT-marked graph marking reachable from the one in Fig. 6

6. CONCLUSIONS

PT-marked graphs without P-vertices (T-vertices) are clearly marked graphs (state machines). This paper showed that unrestricted PT-marked graphs are equivalent to Petri nets with respect to many properties, and elementary vertices may be added or removed without altering any of the properties examined in this paper. After simplifying PT-marked graphs in this manner to the fullest extent, all vertices except for sources and sinks have more than two incident arcs. Hence PT-marked graphs can be used as a reduced model for Petri net analysis, and as a new technique for hierarchically representing systems at various levels of detail. Finally, this paper illustrated that PT-marked graphs are also a promising model for accurate representation of many systems.

As stated in the introduction, a related graph model of Pro-nets has been proposed by Noe [11]. Although P- and T-vertices are subclasses of primitive Pro-net transitions, PT-marked graphs have modeling capability comparable to Pro-nets. Pro-net places are representable by P-vertices, while any Pro-net transition can be replaced by a combination of P- and T-vertices. However, the two models are not identical. For example, the transformation just described may alter the L4 property, and safeness is not imposed by PT-marked graph definitions. PT-marked graphs may be simpler in usage and analysis since they represent logical conjunction and disjunction separately.

REFERENCES

[1] T. Agerwala, "Some applications of Petri nets," Proc. NEC/78, Chicago, 1978.

[2] F. Commoner, Deadlocks in Petri nets, CA-7206-2311, Applied Data Research, Wakefield, MA, June 1972.

[3] For example, N. Deo, Graph Theory with Applications to Engineering and Computer Science, Englewood Cliffs, NJ: Prentice Hall, 1974.

[4] E. Dijkstra, "Co-operating sequential processes," Programming Languages, F. Genuys, ed., New York, NY: Academic Press, 1968.

[5] Y. Han and W. Heimerdinger, Theory of fault tolerance, N00014-75-C-0011, Honeywell Systems & Research Center, Minneapolis, MN, December 1977.

[6] L. Kinney and Y. Han, "Reduction of Petri nets," Proc. of the 14th Allerton Conference on Circuits and Systems Theory, October 1976.

[7] R. Miller, "Relationships among models of parallelism and synchronization," presented at the Conf. on Petri Nets and Related Methods, MIT, July 1975.

[8] T. Murata, "State equation, controllability, and maximal matching of Petri nets," IEEE Trans. on Automatic Control, Vol. AC-22, No. 3, pp. 412-416, June 1977.

[9] T. Murata, "Circuit theoretic analysis and synthesis of marked graphs," IEEE Trans. on Circuits and Systems, Vol. CAS-24, No. 7, pp. 400-405, July 1977.

[10] T. Murata, "Petri nets, marked graphs, and circuit-system theory," Circuits and Systems, Vol. 10, No. 3, pp. 2-12, June 1977.

[11] J. Noe, "Hierarchical modeling with Pro-nets," Proc. NEC/78, Chicago, 1978.

[12] J. Noe and G. Nutt, "Marked E-nets for representation of parallel systems," IEEE Trans. on Comp., Vol. C-22, No. 8, August 1973.

[13] J. Peterson, "Petri nets," Computing Survey, Vol. 9, No. 3, September 1977.

[14] H. Rutishauser, Description of Algol 60, Berlin, Germany: Springer-Verlag, 1967.

AUTOMATION PRINCIPLES FOR THE MANAGEMENT OF THE CRITICALLY ILL

N. DeCLARIS and R ADAMS COWLEY
Maryland Institute for Emergency Medical Services, Baltimore, MD 21201

ABSTRACT

Intensive care is now recognized as a clinical approach to managing the critically ill, patients in life threatening situations. During the last decade, this approach has had phenomenal acceptance in hospitals throughout the world-its value proven beyond any doubt. It is possible to view intensive care as a system involving special manpower, technology resources and computer automated functions. Medical and technological advances promise to make this viewpoint even more effective than in the past, provided a suitable system-theoretic formalism is developed. This paper presents in a concise manner and in 'familiar' system terminology, information and control principle for enhancing the life-saving potential of intensive care through automation.

INTRODUCTION

A casual reader may well wonder, on scanning the titles of the papers of this Allerton Conference, which traditionally deals with fudamental matters of system theory, why this time an apparently medical subject (and a clinical one at that) was included in the program. One of us, N. DeClaris, three years ago in a workshop organized by Prof. A.V. Balakrishnam (which afterward came to be know affectionally as the Catalina Island system theory Review) during a review of future direction in system theory noted that a major challenging problem erea is a phenomenon known by the medical term "shock". For more than a decade the other, R Adams Cowley, has been highly successful in the treatment of shock trauma using a medical system approach, and ultimately founding a unique institution (the first of its kind): the Shock-Trauma Clinic in Baltimore.

System methodology has proven very effective in intensive care medicine. Automation is very much in the mind of those concerned with the management of the critically ill. Those who work in the field recognize that very little is there by way of fundamental system theory to guide their efforts. However, upon reflecting deeper into the problem one appreciates the fact that communication, control and computing considerations are at the heart of any attempt to bring automation in the management of the critically ill. Of course, one may also ask, why bring automation? Well, there are many reasons which we can not take up in this paper although two will suffice. The management of the critically ill (among other) requires painful attention to detail a close and accurate execution of routine tasks. Those are requirements easily met by automation. But if this is all that automation can do for the critically ill a question still remains: How shall it be achieved?

System theorist are accustomed to view instruments, signal processors, and control devices in an abstract setting. But the management of the critically ill involves essential interfaces with patients, physicians and nurses in ways not fully appreciated by persons with limited physiological and medical knowledge. In fact, biomedical system problems involve unique fundamental considerations which can best be formulated in a system-theoretic framework of its own.

CHARACTERIZATION

Critically ill, are by definition, patients:

a. whose life process is sustained by external supports, or

b. who are in acute life threatening situations.

Examples include, patients with multiple trauma following serious automobile accidents and cardiac patients following open heart surgery. The management of the critically ill involves special medical care; its aim being to sustain the very life process through the use of dedicated facilities and specialized methodology. There are thousands of pathological conditions which may change the status of a "normal" healthy" individual to that of an "ill" patient. However, at the present state of medical practice there are only a limited and relatively small number of destructive physiological events which bring about death. Thus the management of the critically ill is aimed at preventing or at least postponing the occurrence of the known mechanisms of death. In system-theoretic terminology then, the life process may be viewed as a trajectory of a biological dynamic system defined in a suitable state-space . That is, the "status of a patient" may be assessed (very primitively) by a minimum number of a set of physiological parameters:

$$x_i \qquad i=1,2,\ldots n \tag{1}$$

which define the physiological state-space Ω. From what is know about x_i, we can safely assume that Ω has all the desirable properties of a Hilbert Space. Points in Ω (the status of a patient at given t) are vectors

$$\underset{\sim}{r} \in \Omega \tag{2}$$

In Ω there exist three proper subspaces

$$\Omega_1 \quad \Omega_2 \quad \Omega_3$$

with the properties that

$$\Omega \supset \Omega_{j+1} \supset \Omega_j \qquad j = 1, 2 \tag{3}$$

and

$$\Omega = \bigcup_i \Omega_i \tag{4}$$

These subspaces have medical significance. Let $\underset{\sim}{r}(t)$ be the trajectory of a specific physiological dynamic system (a patient):

a. if $$\underset{\sim}{r}(t) \in \Omega_N = \Omega_1 \quad \text{for } \forall t \tag{5}$$

then it is "normal",

b. if $$\underset{\sim}{r}(t) \in \Omega_I = \bar{\Omega}_1 \cup \bar{\Omega}_2 \quad \text{for some } t \tag{6}$$

then the system is "ill".

c. if $$\underset{\sim}{r}(t) \in \Omega_x = \bar{\Omega}_2 \cup \bar{\Omega}_3 \quad \text{for some } t \tag{7}$$

then the system is "critically ill".

d. if there exists a t_0 such that

$$\underset{\sim}{r}(t_0) \in \Omega_D = \bar{\Omega}_3 \tag{8}$$

the catastrophic physiological process of "death" has taken place.

What are the system theoretic significance of the subspaces ?

To illuminate this question let us consider the state-space dynamic equilibrium equation which generated the trajectories:

$$\dot{\underset{\sim}{r}} = \underset{\sim}{Q}_N(t; \underset{\sim}{r}, \underset{\sim}{f}, \underset{\sim}{c}) \tag{9}$$

In the above equation we introduced three new vectors:

$\underset{\sim}{Q}_N$ - represents the systemic physiological forces which constraint the trajectory.

$\underset{\sim}{f}$ - represents external (to the patient) forces of system interaction with the normal "natural" enviroment

$\underset{\sim}{c}$ - represents medical observable and/or controllable physiological parameters.

Normally, the physiological dynamic equilibrium involves, in addition to Equ.9, another equation of the type:

$$\underset{\sim}{F}_N\left(t, \frac{d}{dt}; \underset{\sim}{r}, \underset{\sim}{f}, \underset{\sim}{c}\right) = 0 \tag{10}$$

which ensure that

$$\underset{\sim}{r}_{NN} \in \Omega_N$$

When Q_N charges to Q_M, through injury or illness, then the result is

$$\underset{\sim}{r}_{MN} \notin \Omega_N \quad \text{for} \quad \forall t$$

The management of the patient is to apply a "controller":

$$\underset{\sim}{R}_K\left(t, \frac{d}{dt}; \underset{\sim}{r}, \underset{\sim}{f}^d, \underset{\sim}{c}; \underset{\sim}{d}\right) = 0 \tag{11}$$

R_k (most often is applied in step-wise manner and involves a modified vector f^d and a "control vector" d) is chosen so as to force:

$$\underset{\sim}{r}_{MK} \in \Omega_N \quad \text{for} \quad t > t_k \tag{12}$$

Now the distinction between region Ω_I and Ω_X can be made clear. If it is known that a deterministic controller R_k exists (medical management) which ensures that after a finite time t_k Equ.12 is satisfied, even if R_k and d vary by "small amounts", then the trajectory of the system entered only Ω_I; otherwise it entered Ω_X. With other words, once the trajectory enters Ω_X small variations in the controller and control variable may lead to topologically distinct pathways.

By means of the above framework we can also illuminate another medical concept. The boundary between Ω_X and Ω_D is not sharp. In fact if known that there exists a subspace of Ω_X such that, once the trajectory of the system enters it, the only possible continuation following relaxation of the controls is the one which leads to Ω_D. Clearly the aim of intensive care is to avoid by means of suitable "controllers" and "controls" for the trajectory of a critically ill dynamic system to enter this region of "irreversibility". Similarly, if an injured system has already entered the "irreversible" region then costly management schemes should not be undertaken.

At this time medical knowledge of the "irreversible subspace" is, scarce. What then should be the role of automation?

INFORMATION AND CONTROL PRINCIPLES

In the previous section classical control system terminology was used to characterize the basic medical control problem involved in the management of the critically ill. However, as far as practical implementation goes, very little can be accomplished through the direct application of classical control techniques. Even those only casually acquainted with medicine can appreciate the difficulties: while we may know what state-variables characterize the life-process of the complete dynamic system, formulating the dynamic constraints that relate them to the medically observable and controllable variables and to the "enviroment", Q_n is Equ.9, presents a forbidable task. What in fact is done clinically is to view the total biological system as an interconnection of seven vital systems:

1. A Neurological System
2. A Circulatory System
3. A Respiratory System
4. A Fluid-Electrolyte System
5. An Acid-Base Balance System
6. A Metabolic System
7. An Infection Control System.

The identification of these "vital systems" is based on traditional anatomical and function-physiological features. An important system consideration is the fact that if any one of the above seven systems ceases to operate the complete composite system - the life process of the patient - ceases to function. The medical profession, in the last century has developed physicians, specialists who, by virtue of their specialized education and clinical experience, developed skills, methodology and dedicated practice in the management of illness in each of those specific vital systems. It can safely be said that if only one vital system is threatened then the management of that patient is best carried out by a physician specialist and his paramedical personnel and there is little of substance that can be gained by automation - with the exception of record keeping and alarm warnings.

From the system viewpoint it is important to note that in recent years a great deal of quantitative system-theoretic knowledge has accrued for each of the seven vital systems; neurological dynamics are the least understood. Thus with the exception of the neural system, state-variable characterizations for each of the other vital systems have been formulated or are very nearly completed (for the normal case) in the general form of Equ.9 and Equ.10. The key, from the clinical point of view, is the selection of the medically observable and controllable vector $\underset{\sim}{c}$. Thus the components of $\underset{\sim}{c}$ are chosen so as to ensure that there is minimal variation (or it can at least be accounted for) in Equ.10 from patient-to-patient. This is because Equ.9 for any arbitrary $\underset{\sim}{c}$ has very wide variations from patient-to-patient. In the usual case one (or more) models for the vital system is assumed, where the structure remains the same for all patients but individual charactaristics are accounted for by a number of "patient parameters" $\underset{\sim}{S}$; they appear as unknown quantities in the right hand side of Equ.9 The same parameters $\underset{\sim}{S}$ (for the individual patient) also appear in the left-hand of Equ.11 which describes a medical intervention. Thus through appropriately selected values of $\underset{\sim}{d}$ and $\underset{\sim}{f}^{d}$ (in a pre-selected medical protocol R_k) $\underset{\sim}{c}$ is measured. Then from $\underset{\sim}{c}$ "estimates" of $\underset{\sim}{S}$ are obtained. The model is "validated" by still further "medical intervention" which provides "estimates" of the $\underset{\sim}{r}$ vector from measurements of $\underset{\sim}{c}$; the resulting time-services should then satisfy Equ.9 derived from the assumed model in order for it to be valid. A recent example of the effectiveness of this approeach is the use of a measurement (invasive) known as thermo-dilution method to estimate cardiac output: a most usefully medically observable variable used in connection with the study of the circulatory system.

The point made here is that the establishment of a "medical protocol" - management of a patient - involves three aspects simultaneously:

1. parameter estimation
2. model validation and
3. control for stabilization.

Clearly, automation in the management of the ill can enter - in a higher plane that just monitoring and alarm warning - by carrying out complex schemes of parameter estimation and model validation. Microprosessor technology makes this approach economically feasible, although its application in the management of a single vital organ is rarely needed and thus can well be questioned. However, when two or more vital systems are threatened, the patient is generally considered critically ill, and automating these functions on a medical protocol, by means of appropriate instrumentation, algorithmic and computational schemes and physician/nurse utilization is much more attractive at the present time. What is needed are algorithmic formulations of the parameter estimation and model verification schemes of assumed vital system interconnection models to account for known pathophysiological conditions and to utilize current biomedical instrumentation techniques. What should be measured depends heavily on how this measurement will help the physician in his therapeutic task. An excellent example is the new measurement, carried out through the Swan-Ganz catheter, and medically know as the pulmonary wedge pressure, used in the assessment of a physiological model involving both circulatory and pulmonary system aspects. There is one more aspect of automation that we would like to mention.

Following the establishment of automated patient "parameter estimators" and "model validators" it appears that the next step of automation is the use of the algorithmic and computing capability in the clinical ward to develop a single index for evaluating intitatively to what extent a particular patient is critically ill by means of a single index. The use of such and index to assess the effectiveness of a medical protocol, to allocate clinical resources (and even costs) is obvious.

In the previous section we elucidated the crucial role that the "irreversible" subspace plays in the management of the critically ill. The boundary of this region of the state-space (or an estimate based on it) may well provide a measure of the "sickness capacity" of the patient. From this concept it is but a small step to seek for a function of the norm $\|\underline{r}\|$ to serve as a "trauma index".

REFERENCES

1. R Adams Cowley, "Trauma Center, A New Concept for the Delivery of Critical Care" J. of Med. Soc. of New Jersey 74:979. 1977

2. N. DeClaris, "Medical and Engineering Principles of Automated Intensive Care Systems", Invited lecture at the BIOSGMA 78 Conference, Paris, France, April 1978 (to be published).

3. Committee on Pre and Postoperative Care, American College of Surgeons, "Manual of Surgical Intensive Care", W.B. Saunder Co., 1977.

4. V. Klig, "Biomedical Applications of Microprocessors", Por of the IEEE, Feb. 1978

5. N. DeClaris and F. Aldunate, "Microprocessor-Based Modules and Systems for Intensive Care: Proc. Sixth Annual New England Bioengineering Conference, March 1978

6. N. DeClaris, R A. Cowley and B.F. Trump, "A Systems Approach to Intensive Care Medicine" accepted for publication in the American Surgeon, J.B. Lippincott Co., 1978

ON THE REPRESENTATION OF OPTIMAL FIXED-LAG SMOOTHERS FOR CONTINUOUS-TIME RANDOM PROCESSES*

JAMES S. MEDITCH
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

ABSTRACT

Dynamic representations in the form of stable partial differential equations are developed for continuous-time, optimal, fixed-lag smoothing for nonstationary random processes, and previous results for stationary random processes are extended. The need for a new device technology which will render continuous-time, fixed-lag smoothing a viable signal processing technique with wide ranging applications in control and communications is discussed.

1. INTRODUCTION

In estimation theory, the problem of fixed-lag smoothing [1,2] is that of estimating the value of a system state vector or of a message process at a set of time points t-T, with t discrete or continuous and T > 0 and fixed, given noisy measurement information on the state or message process up to t. The quantity T is called the fixed-lag since it is the length of the fixed time interval by which the estimate "lags" the most recent measurement. Interest in this problem stems from the fact that smoothing generally provides significantly smaller estimation error variances than filtering (where T = 0). See, for example, [3, pp. 429-433] in connection with the classical Wiener theory of linear least-squares estimation, [4, Ch. 6] for applications in communication systems, [5] for an application in automatic control, and [6] for bounds in some general cases.

For discrete-time, multivariable, linear dynamic systems with a finite-dimensional state space representation, gaussian white noise disturbances and measurement errors, and gaussian initial conditions, this problem has been solved [7-11]. The results are in the form of discrete-time, sequential algorithms which are stable, easy to implement, and provide the desired fixed-lag smoothed estimate in real time.

In the continuous-time case, the problem has not yielded to solution. The difficulty is known [12] to stem from the presence of an internal, unstable, uncontrollable block in the representation given in [2,13]. One remedy that has been proposed in [14] involves switching between two dynamic representations in an effort to cope with the instability. While this may or may not be workable, it is both theoretically and practically unattractive since the instability is still present and, therefore, a source of difficulty.

Other approaches [15,16] have provided suboptimal solutions either by approximating the system model and specifying an exact filter for this model, or by approximating the optimal smoother itself. Both time and frequency-domain suboptimal solutions have been obtained. Performance relative to the optimal smoother ranges from excellent to fair, and appears to be problem-dependent with only heuristic guidelines available for smoother degree and parameter selection.

---

* Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, under Grant No. AFOSR 78-3546.

We assert that a proper, fundamental attack on this problem must recognize that the optimal, continuous-time, fixed-lag smoother is an infinite-dimensional system, and that the correct framework for stable, exact, dynamic representation resides in the domain of partial differential equations. For stationary random processes, an exact transfer matrix representation which is infinite-dimensional and stable has been obtained [17], and from it has followed the corresponding time-domain dynamic representation [18]. The results in [18] serve to motivate the present work in which we attempt to provide a class of representations for continuous-time, optimal, fixed-lag smoothers with the properties mentioned above. Specifically, our work generalizes that in [18] to nonstationary random processes.

Our presentation is organized as follows. In Section 2, we present the model for the dynamic system whose state we wish to estimate and review some basic results from filtering and smoothing theory. The latter constitute the foundation for our development in Section 3 of the fundamental representation of optimal fixed-lag smoothers. In Section 4, we consider the special case of stationary processes and obtain results which go beyond our earlier work in [18]. Finally, we conclude in Section 5 with a discussion of our results and an indication of some problem areas for future research.

2. MODEL AND BASIC ESTIMATION RELATIONS

We consider systems which are described formally by

$$\dot{x}(t) = F(t)\ x(t) + G(t)\ w(t) \tag{1}$$

$$z(t) = H(t)\ x(t) + v(t) \tag{2}$$

where the dot denotes the derivative with respect to time t with t_o a specified initial time and $t \geq t_o$. Here, $x(\cdot)$ is an n-vector, the state; $w(\cdot)$ an r-vector, the disturbance; $z(\cdot)$ an m-vector, the measurement; and $v(\cdot)$, also an m-vector, the measurement error. The matrices $F(\cdot)$, $G(\cdot)$, and $H(\cdot)$ are continuous and nxn, nxr, and mxn, respectively.

We take $\{w(t),\ t \geq t_o\}$ and $\{v(t),\ t \geq t_o\}$, respectively, to be r-dimensional and m-dimensional zero mean, gaussian, white noise processes with $E[w(t)\ w'(\tau)] = Q(t)\ \delta(t-\tau)$, $E[v(t)\ v'(\tau)] = R(t)\ \delta(t-\tau)$, and $E[v(t)\ w'(\tau)] = 0$ for all $t,\ \tau \geq t_o$ where E denotes the expected value, prime the transpose, and $\delta(\cdot)$ the Dirac delta function. We assume that Q(t) and R(t) are nonnegative definite and positive definite, respectively, for all $t \geq t_o$. We let $x(t_o)$ be zero mean and gaussian with nonnegative definite covariance matrix $P(t_o)$, and assume that it is independent of $\{w(t),\ t \geq t_o\}$ and $\{v(t),\ t \geq t_o\}$.

In the sequel, we shall deal with the two optimal estimates of x(t), $t \geq t_o$, that are specified by

$$\hat{x}(t|t) = E[x(t)|z(\sigma),\ \ t_o \leq \sigma < t]$$

and

$$\hat{x}(t-\tau|t) = E[x(t-\tau)|z(\sigma),\ \ t_o \leq \sigma < t]$$

where $0 \leq \tau \leq T$, T fixed. The first of these is termed the optimal filtered estimate of x(t) and can be generated via the familiar Kalman filter [19-21]. The required relations are

$$\dot{\hat{x}}(t|t) = [F(t) - K(t)\, H(t)]\, \hat{x}(t|t) + K(t)\, z(t)$$

$$= F(t)\, \hat{x}(t|t) + K(t)\, \nu(t) \tag{3}$$

$$\nu(t) = z(t) - H(t)\, \hat{x}(t|t) \tag{4}$$

$$K(t) = P(t|t)\, H'(t)\, R^{-1}(t) \tag{5}$$

$$\dot{P}(t|t) = F(t)\, P(t|t) + P(t|t)\, F'(t) - P(t|t)\, H'(t)\, R^{-1}(t)\, H(t)\, P(t|t)$$

$$+ G(t)\, Q(t)\, G'(t) \tag{6}$$

subject to the initial conditions $\hat{x}(t_o|t_o) = 0$ and $P(t_o|t_o) = P(t_o)$. In these relations, $(\cdot)^{-1}$ denotes the matrix inverse, $P(t|t)$ is the covariance matrix of the filtering error $\tilde{x}(t|t) \triangleq x(t) - \hat{x}(t|t)$, and $\{\nu(t),\ t \geq t_o\}$ is the innovations process which is known [21] to be (i) a zero mean, gaussian, white noise with covariance matrix $R(t)$, and (ii) causally equivalent to $\{z(t),\ t \geq t_o\}$.

In order that our results make sense, we assume that the system description $[F(t), G(t), H(t), Q(t), R(t),\ t \geq t_o]$ satisfies the sufficient conditions of [22] which guarantee the asymptotic stability of the Kalman filter in Eqs. (3) through (6).

For fixed values of $\tau \in (0,T]$, the second estimate above is called a fixed-lag smoothed estimate of $x(t)$. It is given [13] by

$$\hat{x}(t-\tau|t) = \hat{x}(t-\tau|t-\tau) + \int_{t-\tau}^{t} P(t-\tau,s)\, H'(s)\, R^{-1}(s)\, \nu(s)\, ds \tag{7}$$

where $t \geq t_o$. In Eq. (7), $P(\cdot,\cdot)$ is the covariance kernel of the filtering error and is defined by

$$P(t,s) \triangleq E[\tilde{x}(t|t)\, \tilde{x}'(s|s)] \qquad t,s \geq t_o$$

It is well-known [13] that

$$P(t,s) = \begin{cases} \Phi(t,s)\, P(s|s) & t \geq s \\ P(t|t)\, \Phi'(s,t) & s \geq t \end{cases} \tag{8}$$

where

$$\frac{\partial\, \Phi(t,s)}{\partial\, t} = [F(t) - K(t)\, H(t)]\, \Phi(t,s)$$

and (9)

$$\frac{\partial\, \Phi(s,t)}{\partial\, t} = -\, \Phi(s,t)\, [F(t) - K(t)\, H(t)]'$$

with $\Phi(s,s) = I$, the nxn identity matrix, for all s.

The corresponding fixed-lag smoothing error covariance matrix

$$P(t-\tau|t) \triangleq E[\tilde{x}(t-\tau|t)\, \tilde{x}'(t-\tau|t)]$$

is [13]

$$P(t-\tau|t) = P(t-\tau|t-\tau) - \int_{t-\tau}^{t} P(t-\tau,s)\ H'(s)\ R^{-1}(s)\ H(s)\ P(s,t-\tau)\ ds \qquad (10)$$

for each $\tau \; \varepsilon \; (0,T]$.

Given these basic results, we are now in a position to develop the desired dynamical representation for the fixed-lag smoother and its associated error covariance matrix.

3. FIXED-LAG SMOOTHER REPRESENTATION

For any $t \geq t_o$ and $\tau \; \varepsilon \; [0,T]$, we have via partial differentiation in Eq. (7) that

$$\frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ t} = \frac{\partial\ \hat{x}(t-\tau|t-\tau)}{\partial\ t} + \int_{t-\tau}^{t} \frac{\partial\ P(t-\tau,s)}{\partial\ t}\ H'(s)\ R^{-1}(s)\ \nu(s)\ ds$$

$$+ P(t-\tau,t)\ H'(t)\ R^{-1}(t)\ \nu(t)$$

$$- P(t-\tau|t-\tau)\ H'(t-\tau)\ R^{-1}(t-\tau)\ \nu(t-\tau) \qquad (11)$$

and that

$$\frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ \tau} = \frac{\partial\ \hat{x}(t-\tau|t-\tau)}{\partial\ \tau} + \int_{t-\tau}^{t} \frac{\partial\ P(t-\tau,s)}{\partial\ \tau}\ H'(s)\ R^{-1}(s)\ \nu(s)\ ds$$

$$+ P(t-\tau|t-\tau)\ H'(t-\tau)\ R^{-1}(t-\tau)\ \nu(t-\tau) \qquad (12)$$

But from the fact that

$$\frac{\partial\ \hat{x}(t-\tau|t-\tau)}{\partial\ t} = - \frac{\partial\ \hat{x}(t-\tau|t-\tau)}{\partial\ t} \quad \text{and} \quad \frac{\partial\ P(t-\tau,s)}{\partial\ t} = - \frac{\partial\ P(t-\tau,s)}{\partial\ \tau}$$

Eq. (11) becomes

$$\frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ t} = - \frac{\partial\ \hat{x}(t-\tau)t-\tau)}{\partial\ \tau} - \int_{t-\tau}^{t} \frac{\partial\ P(t-\tau,s)}{\partial\ \tau}\ H'(s)\ R^{-1}(s)\ \nu(s)\ ds$$

$$+ P(t-\tau,t)\ H'(t)\ R^{-1}(t)\ \nu(t)$$

$$- P(t-\tau|t-\tau)\ H'(t-\tau)\ R^{-1}(t-\tau)\ \nu(t-\tau) \qquad (13)$$

Noting that the first, second and last terms on the right hand side in Eq. (13) are identically the negative of the right-hand side in Eq. (12), we have immediately the fixed-lag smoother representation

$$\frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ t} = - \frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ \tau} + P(t-\tau,t)\ H'(t)\ R^{-1}(t)\ \nu(t) \qquad (14)$$

where $t \geq t_o$ and $0 \geq \tau \geq T$.

Utilizing the second relation in Eq. (8), we get

$$\frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ t} = \frac{\partial\ \hat{x}(t-\tau|t)}{\partial\ \tau} + M(t,\tau)\ H'(t)\ R^{-1}(t)\ \nu(t) \qquad (15)$$

where

$$M(t,\tau) \triangleq P(t-\tau|t-\tau)\ \Phi'(t,t-\tau) \qquad (16)$$

with $\Phi(t,t-\tau)$ coming from the first relation in Eq. (9).

The initial condition for Eq. (15) is evidently the function $x(t_o-\tau|t_o) = 0$ over $0 < \tau \leq T$, and the boundary condition (at $\tau = 0$) is $\hat{x}(t|t)$ which is obtained from the Kalman filter, Eqs. (3) through (6), for all $t \geq t_o$.

Since the estimate of interest here is the one for the specified lag T, we define the output relation associated with the system in Eq. (15) to be $y(t) = \hat{x}(t-T|t)$ for $t \geq t_o + T$.

To determine the corresponding dynamic representation for the fixed-lag smoothing error covariance, we proceed identically as above via operations on Eq. (10). We differentiate separately first with respect to t and second with resepct to τ. Then, noting that

$$\frac{\partial\ P(t-\tau,s)}{\partial\ t} = -\frac{\partial\ P(t-\tau,s)}{\partial\ \tau}$$

and similarly for $P(s,t-\tau)$, and again utilizing Eqs. (8) and (9), we are led to the result

$$\frac{\partial\ P(t-\tau|t)}{\partial\ t} = -\frac{\partial\ P(t-\tau|t)}{\partial\ \tau} - M(t,\tau)\ H'(t)\ R^{-1}(t)\ H(t)\ M'(t,\tau) \qquad (17)$$

where $t \geq t_o$, $0 \leq \tau \leq T$, and $M(t,\tau)$ is given by Eq. (16). The initial condition for Eq. (17) is the function $P(t_o-\tau|t_o) = 0$ over $0 < \tau \leq T$, and the boundary condition (at $\tau = 0$) is $P(t|t)$, $t \geq t_o$, which is the solution of Eq. (6).

In parallel with the output equation for the fixed-lag smoother, we take the output relation for Eq. (17) to be $\Sigma(t) = P(t-T|t)$ where $t \geq t_o + T$.

The fixed-lag smoother representation that we have given in Eqs. (15), (16), and (17) is mathematically equivalent to, but structurally much simpler than, the earlier one that is due to Kwakernaak [23]. In particular, all relations in our representation are linear whereas his involves a matrix Riccati partial differential equation in three arguments. The latter must be solved in order to determine the weighting function, equivalently our $M(t,\tau)$, that appears in the partial differential equation for the fixed-lag smoother, and also to evaluate the fixed-lag smoothing error covariance matrix. By way of comparison, our characterization of $M(t,\tau)$ is direct. Further, our error covariance relation, Eq. (17), is not coupled to our smoother realization, Eq. (15), whereas the two are coupled in Kwakernaak's formulation. As a result, we can implement the smoother without the error covariance equation; this is not possible with the representation in [23].

In concluding this section, we note that the input to our distributed-parameter representation is the innovations process as a consequence of our starting point, Eq. (7). An alternative and equivalent starting point could have been the integral relationship in [24] in which the filtered estimate $\hat{x}(\cdot|\cdot)$ is the input. The corresponding dynamic representation has been obtained in [18] for stationary random processes. We defer the non-stationary case to a future study, but take note of the observation made in [16] in connection with frequency-domain approximations that $\hat{x}(\cdot|\cdot)$ may be a preferred input since it is bandlimited whereas $\nu(\cdot)$ is a white noise.

4. STATIONARY PROCESSES

As a special case, we let $t_0 \to -\infty$ and consider the situation where $\{x(t), t > -\infty\}$ and $\{z(t), t > -\infty\}$ are stationary random processes. In particular, we assume that F is constant and has eigenvalues in the open left-half plane; that G, H, Q, and R are also constant; and that P is the unique nonnegative definite solution of $FP + FP' - PH' R^{-1} HP + GQG' = 0$. Under these conditions, we have from Eq. (16) and the first expression in Eq. (8) that

$$M(t,\tau) = P\, e^{(F-KH)'\tau}$$

As a result, Eqs. (15) and (17) become

$$\frac{\partial\, \hat{x}(t-\tau|t)}{\partial\, t} = - \frac{\partial\, \hat{x}(t-\tau|t)}{\partial\, \tau} + P\, e^{(F-KH)'\tau}\, H'\, R^{-1}\, \nu(t) \tag{18}$$

and

$$\frac{\partial\, P(t-\tau|t)}{\partial\, t} = - \frac{\partial\, P(t-\tau|t)}{\partial\, \tau} + P\, e^{(F-KH)'\tau}\, H'\, R^{-1}\, H\, e^{(F-KH)\tau}\, P \tag{19}$$

respectively, where $t > -\infty$. The boundary conditions for Eq. (18) is $\hat{x}(t|t)$, $t > -\infty$, from the Kalman filter, while that for Eq. (19) is $P(t|t) = P$, $t > -\infty$. The corresponding desired outputs are, of course, $y(t) = \hat{x}(t-T|t)$ and $\Sigma(t) = P(t-T|t)$ as before.

As an alternative to the above representation, we obtain below a dynamical one which is based on the formulation used in [16] to develop suboptimal smoothers. The representation here is, however, exact.

We have from Eq. (7) and stationarity that

$$\hat{x}(t-T|t) = \hat{x}(t-T|t-T) + \int_{t-T}^{t} P(t-T-s)\, H'\, R^{-1}\, \nu(s)\, ds$$

where $t > -\infty$ and, from Eqs. (8) and (9), that

$$P(t-T-s) = P\, e^{(F-KH)'(s-t+T)}$$

for $t-T \le s \le t$. Via the change of variable $s = t - \sigma$, the above result becomes

$$\hat{x}(t-T|t) = \hat{x}(t-T|t-T) + \int_{0}^{T} P\, e^{(F-KH)'(T-\sigma)}\, H'\, R^{-1}\, \nu(t-\sigma)\, d\sigma \tag{20}$$

The interpretation [16] of Eq. (20) is that the fixed-lag smoother can be realized by a Kalman filter, for which current time is t, and a delay line to obtain $\hat{x}(t-T|t-T)$, and by passing $\nu(\cdot)$, also available from the Kalman filter, through a finite-impulse-response (FIR) filter whose impulse response is

$$W(\sigma) = \begin{cases} P\, e^{(F-KH)'(T-\sigma)} & 0 \le \sigma \le T \\ 0 & \text{elsewhere} \end{cases} \tag{21}$$

We obtain the exact dynamic representation for $W(\cdot)$ by considering

$$\theta(t,\tau) = \int_{t-\tau}^{t} P(t-\tau-s)\, H'\, R^{-1}\, \nu(s)\, ds$$

where $t > -\infty$ and $0 \le \tau \le T$. Then, proceeding exactly as in Section 3, we are led to the result

$$\frac{\partial\, \theta(t,\tau)}{\partial\, t} = - \frac{\partial\, \theta(t,\tau)}{\partial\, \tau} + P\, e^{(F-KH)'\tau}\, H'\, R^{-1}\, \nu(t) \tag{22}$$

The initial condition for Eq. (22) is $\theta(-\infty,\tau) = 0$ where $0 \le \tau \le T$, and the boundary condition is $\theta(t,t) = 0$, $t > -\infty$. In this case, Eq. (20) becomes $\hat{x}(t-T|t) = \hat{x}(t-T|t-T) + \theta(t,T)$.

We note that the same representation can also be obtained for the non-stationary case treated in Section 3 with only minor changes.

We turn now to the question of the steady-state ($t \to \infty$) fixed-lag smoothing error covariance matrix. Under the conditions that led to the representation in Eq. (18), Eq. (10) becomes

$$P(t-T|t) = P - \int_{t-T}^{t} P(t-T-s)\, H'\, R^{-1}\, H\, P(s-t+T)\, ds \tag{23}$$

where as a consequence of Eqs. (8) and (9)

$$P(t-T-S) = P\, e^{(F-KH)'(s-t+T)}$$

and (24)

$$P(s-t+T) = e^{(F-KH)(s-t+T)}\, P$$

Since the Kalman filter is asymptotically stable, the eigenvalues of (F-KH) are in the open left-half plane. Hence, it follows [25] that

$$\lim_{t \to \infty} P(t-T|t) \triangleq \Sigma$$

exists and is finite.

Then, differentiating in Eq. (23) when t is arbitrarily large, we have with the aid of Eq. (24) that

$$0 = - P e^{(F-KH)'T} H' R^{-1} H e^{(F-KH)T} + P H' R^{-1} H P$$

$$-P(F-KH)' \int_{t-T}^{t} e^{(F-KH)'(s-t+T)} H' R^{-1} H e^{(F-KH)(s-t+T)} ds P$$

$$- P \int_{t-T}^{t} e^{(F-KH)'(s-t+T)} H' R^{-1} H e^{(F-KH)(s-t+T)} ds (F-KH)$$

Utilizing the fact that*

$$(F-KH)' = - P^{-1}(F-GQG'P^{-1}) P$$

(see, for example [17]), the expression above becomes

$$0 = -P e^{(F-KH)'T} H' R^{-1} H e^{(F-KH)T} + P H' R^{-1} H P$$

$$+ (F+GQG'P^{-1}) \int_{t-T}^{t} P e^{(F-KH)'T} H' R^{-1} H e^{(F-KH)T} P ds$$

$$+ \int_{t-T}^{t} P e^{(F-KH)'(s-t+T)} H' R^{-1} H e^{(F-KH)(s-t+T)} P ds(F+GQG'P^{-1})'$$

But from Eqs. (23) and (24), each of the integrals here is simply P - Σ. Hence, we have

$$0 = - P \Phi'(T) H R^{-1} H \Phi(T) P + (F+GQG'P^{-1}) \Sigma + \Sigma(F+GQG'P^{-1})'$$
$$- G Q G' - (FP+PF'-PH'R^{-1}HP+GQG') \tag{25}$$

where $\Phi(T) \triangleq e^{(F-KH)T} P$. Finally as is well-known for the stationary Kalman filter, the last term in parentheses in Eq. (25) is identically zero. Therefore, we are led to the system of linear equations

$$(F+GQG'P^{-1}) \Sigma + \Sigma(F+GQG'P^{-1})' = P\Phi'(T)H'R^{-1}H\Phi(T)P + GQG' \tag{26}$$

whose solution Σ is the desired steady-state fixed-lag smoothing error covariance matrix. The need for P^{-1} in Eq. (26) can be met [13,20] by solving $F'U + UF + UGQG'U - H'R^{-1}H = 0$ where $U = P^{-1}$ rather than by numerical inversion of P.

We remark that Eqs. (18) and (26) were first developed in [18] and [1], respectively. However, the approach used here is different, and the other results in this section serve to provide a more complete picture for the stationary case. Also, it should be noted that Eq. (26) is valid for any fixed-lag smoother representation that is equivalent to the one in Eq. [18].

---

* Only cases where P is nonsingular are of interest here. Otherwise, at least one linear functional of x(t) is known exactly from the Kalman filter and smoothing is not needed to estimate such functionals.

5. CONCLUSION

In this paper, we have developed the basic dynamic representation for continuous-time, optimal, fixed-lag smoothing, our principle results being Eqs. (15) and (17). We have also investigated the implications of our results in the case of stationary random processes.

Contemporary technology does not provide the distributed-parameter devices necessary to realize the operations called for by the representations in Eqs. (15), (18), and (22). In particular, we have in mind here wave type devices rather than digital ones whose behavior is characterized by say Eq. (15). Clearly, the theory is well ahead of practice, but provides the details of the types of operations of which devices must be capable. The burden is thus upon device technology to provide the means for rendering continuous-time, optimal, fixed-lag smoothing a viable signal processing technique. There is no doubt that control and communication system applications will follow immediately thereafter.

REFERENCES

1. J. S. Meditch, "Orthogonal projection and discrete optimal linear smoothing," SIAM J. Contr., vol. 6, no. 1, pp. 74-89, Feb. 1967.

2. __________, "On optimal linear smoothing theory," Inform. and Contr., vol. 10, no. 6, pp. 598-615, June 1967.

3. Y. W. Lee, Statistical Communication Theory. New York: Wiley, 1960.

4. A. J. Viterbi, Principles of Coherent Communication. New York: McGraw-Hill, 1966.

5. A. Lindquist, "On optimal stochastic control with smoothed information," Inform. Sci., vol. 1, no. 1, pp. 55-85, Dec. 1968.

6. B. D. O. Anderson and S. Chirarattanon, "Smoothing as an improvement on filtering: a universal bound," Electron. Letters, vol. 7, no. 8, pp. 524-525, Sept. 9, 1971.

7. H. E. Rauch, "Solutions to the linear smoothing problem," IEEE Trans. Automat. Contr., vol. AC-8, no. 4, pp. 371-372, Oct. 1963.

8. K. K. Biswas and A. K. Mahalanabis, "Application of state augmentation approach to problems of optimal smoothing, " Dept. of Elec. Engr., Indian Inst. of Technology, New Delhi, India, Feb. 1971.

9. S. Chirarattananon and B. D. O. Anderson, "The fixed-lag smoother as a stable finite-dimensional linear system," Automatica, vol. 7, no. 6, pp. 657-669, Nov. 1971.

10. J. B. Moore, "Discrete-time fixed-lag smoothing algorithms," Automatica, vol. 9, no. 2, pp. 163-173, Mar. 1973.

11. K. K. Biswas and A. K. Mahalanabis, "A stable fixed-lag smoothing algorithm," Automatica, vol. 9, no. 3, pp. 393-397, May 1973.

12. C. N. Kelly and B. D. O. Anderson, "On the stability of fixed-lag smoothing algorithms," J. Franklin Inst., vol. 291, no. 4, pp. 271-281, Apr. 1971.

13. T. Kailath and P. Frost, "An innovations approach to least-squares estimation, Part II: linear smoothing in additive white noise," IEEE Trans. Automat. Contr., vol. AC-13, no. 6, pp. 655-660, Dec. 1968.

14. P. K. S. Tam and J. B. Moore, "Stable realization of fixed-lag smoothing equations for continuous-time signals," IEEE Trans. Automat. Contr., vol. AC-19, no. 1, pp. 84-87, Feb. 1974.

15. S. Chirarattananon and B. D. O. Anderson, "Stable fixed-lag smoothing of continuous-time processes," IEEE Trans. Info. Th., vol. IT-20, no. 1, pp. 25-36, Jan. 1974.

16. B. D. O. Anderson and H. B. Doan, "Design procedure for stable suboptimal fixed-lag smoothers," IEEE Trans. Automat. Contr., vol. AC-22, no. 6, pp. 949-952, Dec. 1977.

17. J. S. Meditch, "Stability in fixed-lag smoothing," Proc. 1972 Joint Automat. Contr. Conf., Stanford, Calif., Aug. 1972, pp. 838-841.

18. __________, "Stable continuous-time optimal fixed-lag smoothers for stationary random processes," IEEE Trans. Automat. Contr., to appear.

19. R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," Trans. ASME, Ser. D, J. Basic Engr., vol. 83, no. 1, pp. 95-108, Mar. 1961.

20. J. S. Meditch, Stochastic Optimal Linear Estimation and Control. New York: McGraw-Hill, 1969.

21. T. Kailath, "An innovations approach to least-squares estimation, Part I: linear filtering in additive white noise," IEEE Trans. Automat. Contr., vol. AC-13, no. 6, pp. 646-655, Dec. 1968.

22. B. D. O. Anderson, Stability properties of Kalman-Bucy filters," J. Franklin Inst., vol. 291, no. 2, pp. 137-144, Feb. 1971.

23. H. Kwakernaak, "Optimal filtering in linear systems with time delays," IEEE Trans. Automat. Contr., vol. AC-12, no. 2, pp. 169-173, Apr. 1967.

24. B. D. O. Anderson and S. Chirarattananon, "New linear smoothing formulas," IEEE Trans. Automat. Contr., vol. AC-17, no. 1, pp. 160-161, Feb. 1972.

25. R. Bellman, Introduction to Matrix Analysis. New York: McGraw-Hill, 1960.

ESTIMATION OF THE ORBITAL STATES OF A SATELLITE FROM GROUND-BASED MEASUREMENTS

S.A. AZIM and N.K. SINHA
Group on Simulation, Optimization and Control
Faculty of Engineering
McMaster University
Hamilton, Ontario, Canada L8S 4L7

ABSTRACT

A number of different approaches have been considered for estimating the orbital states of a synchronous-orbit satellite from ground-based measurements taking into account the nonlinear nature of the state and observation models. The following methods are compared using actual observations: (i) the extended Kalman filter, (ii) invariant embedding, (iii) innovations and (iv) quasilinearization.

INTRODUCTION

It is necessary to estimate accurately the orbital states of a satellite from ground-based measurements, which are contaminated with noise in order that corrective action may be taken to maintain the satellite in a desired orbit. The problem is complicated by the fact that the state and output equations are nonlinear and also the noise statistics are not known a priori.

This paper examines three distinct methods for the recursive estimation of the state variables of a continuous time nonlinear plant on the basis of measuring discrete-time outputs of the plant in the presence of noise.

The three approaches for nonlinear estimation are (i) the Invariant Imbedding approach, (ii) the Innovations approach, and (iii) the Quasilinearization approach.

These algorithms are applied to the estimation of the states of a synchronous orbit satellite and compared to the extended Kalman filter algorithm presented in an earlier paper [1].

A NONLINEAR FILTER WITH SECOND-ORDER WEIGHTING FUNCTIONS VIA INVARIANT IMBEDDING

In engineering and physical sciences there occur many two-point or multipoint boundary-value problems. Since these problems usually are nonlinear, they are accompanied by various analytical and numerical difficulties. The numerical difficulties are caused by the fact that not all the conditions are given at one point. Furthermore for a large number of problems (i.e. orbit determination) the starting or guessed missing condition must be very close to the correct and yet unknown condition before the procedure will converge.

The invariant imbedding approach is a useful technique for solving such problems [2]. It is used to derive some useful results in nonlinear filtering and state estimation problems. A sequential estimation scheme was obtained [3] for scalar system. A generalization for the multivariable system is presented in this paper with the application to the orbit determination problem. The main advantage of this algorithm is that no statistical assumptions are made concerning the noise or disturbances. The commonly used least-squares criterion is employed to obtain optimal estimates.

Consider a system represented by the nonlinear differential equation

$$\frac{dx}{dt} = f(x,t)$$

The observed state is given by

$$z(t) = h(x,t) + (\text{meas. or observation errors})$$

On the basis of this observed signal z(t) in the interval $0 \leq t \leq t_f$ the problem is to estimate the current state $x(t_f)$, so that the following integral is minimized:

$$J = \int_{t_o}^{t_f} (x(t) - z(t))^2 \, dt$$

or estimate the unknown condition

$$x(t_f) = c$$

so that J is minimized.

Define the new variable

$$y(t) = \int_{o}^{t} (x(t) - z(t))^2 \, dt$$

$$y(t_f) = c$$

Consider now the general problem

$$\frac{dx}{dt} = f(x,t)$$

$$\frac{dy}{dt} = (x(t) - z(t))^2$$

with the given condition

$$x(a) = c$$

with $0 \leq t \leq a$. In other words, the missing final condition $y(t_f)$ is to be obtained by considering a family of processes with different final points a.

Define $r(c,a) = y(a)$

By invariant imbedding an expression for r in terms of c and a is obtained and called the "Invariant Imbedding equation"

$$f(c,a) \frac{\partial r(c,a)}{\partial c} + \frac{\partial r(c,a)}{\partial a} = (c - z(a))^2$$

The derivation is in reference [2] and is based on Taylor's series expansion up to the second order.

The problem now is to obtain a series of values of c which minimize the cost function r(c,a) for a series of final terminal points, a. Let these minimizing values be denoted by e(a) which are the optimal estimate of x, so

$$\frac{\partial r(e,a)}{\partial c} = r_c(e,a) = 0$$

After several manipulations we obtain

$$\frac{de}{da} = f(e,a) + q(a)\ (z(a) - e)$$

where

$$q(a) = \frac{2}{r_{cc}(e,a)}$$

This equation gives the optimal estimate, the function q(a) is considered as a weighting function and should be obtained. The details of the derivations are in reference [2] and the result is given by

$$\frac{dq}{da} = 2\ f_c(e,a)\ q(a) - q^2(a)$$

The above results can be generalized to systems with dynamics represented by M differential equations i.e. x and f are M-dimensional vectors. z and h are m-dimensional vectors. In this case we obtain the filter equations

$$\frac{de}{dt} = f(e,t) + q(t)\ h_e^T(e,t)\ [z(t) - h(e,t)]$$

$$\frac{dq}{dt} = f_e(e,t)\ q(t) + q(t)\ [f_e(e,t)]^T$$

$$+ q(t)\ \{h_{ee}(e,t)\ [z(t) - h(e,t)\}\ q(t)$$

$$- q(t)\ [h_e(e,t)]^T\ h_e(e,t)\ q(t)$$

where

$$f_e = \begin{bmatrix} \frac{\partial f_1}{\partial e_1} & \cdots & \frac{\partial f_1}{\partial e_M} \\ \vdots & & \\ \frac{\partial f_M}{\partial e_1} & \cdots & \frac{\partial f_M}{\partial e_M} \end{bmatrix}$$

$$h_e = \begin{bmatrix} \frac{\partial h_1}{\partial e_1} & \cdots & \frac{\partial h_1}{\partial e_M} \\ \vdots & & \\ \frac{\partial h_m}{\partial e_1} & \cdots & \frac{\partial h_m}{\partial e_M} \end{bmatrix}$$

$$h_{ee}[z-h] = \begin{bmatrix} h^T_{e_1e_1}(z-h) & h^T_{e_1e_2}(z-h) & \cdots & h^T_{e_1e_M}(z-h) \\ \vdots & & & \\ h^T_{e_Me_1}(z-h) & \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot & & h^T_{e_Me_M}(z-h) \end{bmatrix}$$

For practical applications we consider $e(t)$ = state estimate $\hat{x}(t)$ and $q(t)$ = covariance of the estimation error.

So the continuous time second-order filter based on the invariant imbedding approach is given by

$$\frac{d\hat{x}}{dt} = f(\hat{x},t) + q(t)\ h^T_x(\hat{x},t)\ [z(t) - h(\hat{x},t)]$$

$$\frac{dq}{dt} = f_x(\hat{x},t)\ q(t) + q(t)\ [f_x(\hat{x},t)]^T$$

$$+\ q(t)\ \{h_{xx}(\hat{x},t)\ [z(t) - h(\hat{x},t)\}\ q(t)$$

$$-\ q(t)\ [h_x(\hat{x},t)]^T\ h_x(\hat{x},t)\ q(t)$$

THE INNOVATIONS APPROACH

Kailath has provided considerable insight into the problem of optimum filtering and estimtaion by proposing and proving the "Innovations Theorem" [4], [5]. Using this theorem Kailath has been able not only to derive the filtering equations of Kalman and Bucy in a simpler and more elegant manner, but also to extend the concepts to certain nonlinear problems.

A stochastic approximation technique proposed by Ibidapo-Obe [6] is utilized for numerical computation of the innovations processes and consequently the system state.

Consider the general system

$$\frac{dx}{dt} = f(x,t) + w(t)$$

with observation model given by

$$z(t) = h(x,t) + v(t)$$

It is well known that the optimal estimate of x_t denoted by $\hat{x}_{t|\tau}$ is given by

$$\hat{x}_{t|\tau} = E\{x_t \mid z_s,\ 0 \leq s \leq \tau\}$$

Define the innovations process as

$$\nu_t = z_t - \hat{y}\ t|t$$

where

$$y(t) = h(x,t)$$

and

$$\hat{y}_{t|t} = h(\hat{x}_s,\ s \leq t)$$

The innovations approach transforms the data process y(·) into a white noise process ν(·) called the innovations process. The optimal estimate is determined as a functional of the innovations process

$$\hat{x}_{t|\tau} = E\{x_t \mid \nu_s,\ 0 \leq s \leq \tau\}$$

The differential form for the state filtered estimate is derived in [6] and given by

$$\frac{d\hat{x}}{dt} = \hat{f}(t,x_t) + \hat{K}(t,x_t)\ \nu_t$$

where

$$\hat{K}(t,x_t) = E\{x_t\ \nu_t^T \mid \nu_\sigma,\ 0 \leq \sigma < t\}$$

This equation is simply a representation for the recursive filtering rather than an explicit formula since the functions $\hat{f}(\cdot,\cdot)$ and $\hat{K}(\cdot,\cdot)$ are in general indeterminate.

Stochastic approximation is one of those methods which are not necessarily optimal in any sense but yield recursive estimators possessing well-defined convergence properties.

The algorithm proposed in [6] is the iterative formula

$$K_t^{i+1} = K_t^i + \gamma_t^i f(\nu_t^i)$$

where

$$\gamma_t^i = \frac{a}{t} \quad \text{or} \quad \frac{a}{b+t} \qquad a,b > 0$$

$f(\nu_t^i)$ = function of the innovations.

The function $f(\nu_i)$ should be a null matrix when K_i is identical with K_{opt}. There are several choices for $f(\nu_i)$ to satisfy this condition. In this paper we utilize the orthogonality between the innovations and the one step predicted state.

The following stochastic approximation algorithm is used

$$K_{i+1} = K_i + \frac{1}{i+1} \frac{\hat{x}_{i+1} \nu'_{i+1}}{||\hat{x}_{i+1} \nu'_{i+1}||^2}$$

Proof of convergence is given in [7].

SECOND-ORDER NONLINEAR FILTER BASED ON QUASILINEARIZATION

A sequential "second-order" nonlinear filter which generates estimates for the state-variables of a continuous time nonlinear dynamical system from discrete noisy measurements was developed by Athans, Wishner and Bertolini [8]. In this approach state and output nonlinearities are taken into account by simply retaining second-order terms in the usual Taylor series expansions. The detailed derivations of the filter are given in reference [8]. The summary of the results is presented below

Plant

$$\dot{x}(t) = f(x(t)); \; x(t_o) = x_o \quad \text{(state equation)}$$

$$y(t) = h(x(t)) \quad \text{(output equation)}$$

Observations

$$z_k = h(x_k) + v_k$$

State estimate at Observation time

$$t_k : \hat{x}_k$$

$$\hat{e}_k = x_k - \hat{x}_k$$

$$\Sigma_k = E\{\hat{e}_k \hat{e}'_k\}$$

State estimate at t, $t_k < t < t_{k+1}$: $w(t)$

$$e(t) = x(t) - w(t)$$

$$S(t) = E\{e(t)\, e'(t)\}$$

Basic Assumption

$\hat{e}_k$ and $e(t)$ Gaussian, zero-mean

Starting Conditions

$$\hat{x}_o = w(t_o) = E\{x_o\} = \bar{x}_o$$

$$\Sigma_o = S(t_o) = E\{(x_o - \bar{x}_o)(x_o - \bar{x}_o)'\}$$

Continuous time Filter: $t_k < t < t_{k+1}$

$$\dot{w}(t) = f(w(t)) + \frac{1}{2} \sum_{i=1}^{n} Q_i \operatorname{tr}[F_i(w(t))\, S(t)]$$

$$w(t_k) = \hat{x}_k$$

$$\dot{S}(t) = A(w(t))\, S(t) + S(t)\, A'(w(t))$$

$$S(t_k) = \Sigma_k$$

Update at $t = t_{k+1}$

$$\hat{x}_{k+1} = w_{k+1} + G_{k+1} [z_{k+1} - h(w_{k+1})] - \pi_{k+1}$$

$$\pi_{k+1} = 1/2\, G_{k+1} \sum_{j=1}^{m} \phi_j \operatorname{tr}[D_j(w_{k+1})\, S_{k+1}]$$

$$G_{k+1} = S_{k+1}\, C'(w_{k+1}) [C(w_{k+1})\, S_{k+1}\, C'(w_{k+1}) + R_{k+1} + L_{k+1}]^{-1}$$

$$\Sigma_{k+1} = S_{k+1} - S_{k+1}\, C'(w_{k+1}) [C(w_{k+1})\, S_{k+1}\, C'(w_{k+1}) + R_{k+1} + L_{k+1}]^{-1} C(w_{k+1})\, S_{k+1}$$

$$(L_{k+1})_{ij} = \frac{1}{2} \operatorname{tr}[D_i(w_{k+1})\, S_{k+1}\, D_j(w_{k+1})\, S_{k+1}]$$

where $(L_{k+1})_{ij}$ is the ij^th element of the mxm matrix L_{k+1}.

RESULTS OF SIMULATION

These three algorithms are tested and compared with the adaptive filtering algorithm using the extended Kalman filter [1]. The comparison is made using the data supplied by the Space-craft Mechanics Division of the Communications Research Centre in Ottawa, obtained from actual records of observations.

The dynamic state model used is given in detail in reference [9], it consists of seven state variables which define the velocity and the rotation tranformation for the coordinate frame triad.

$$\underline{x} = [C, R_1, R_2, e_1, e_2, e_3, e_4]$$

(C, R_1, R_2) define the velocity state, (e_1, e_2, e_3, e_4) are the Euler parameters.

The observations are the elevation, range and azimuth

$$\underline{y} = (\gamma, h, A)$$

The nonlinear state and observation models are given in [9].

Data covering 24 hours are simulated using a CDC 6400 computer.

The performance of the four filters are compared on the basis of the trace of the error covariance matrix

$$\text{trace } [P] = \sum_{i=1}^{7} p_{ii}$$

We start with initial states (x_o) supplied by the CRC

$$C = 3.0766 \text{ km/sec}$$

$$R_1 = R_2 = 0 \text{ km/sec}$$

$$e_1 = \sin i/2 \cos (r-v)/2$$

$$e_2 = \sin i/2 \sin (r-v)/2$$

$$e_3 = \cos i/2 \sin (r+v)/2$$

$$e_4 = \cos i/2 \cos (r+v)/2$$

where $i = 0.9^\circ$ $r = 240^\circ$ $v = -r$

and initial observations are

$$\gamma = 24.6^\circ$$

$$h = 39180 \text{ km}$$

$$A = 229.8^\circ$$

The initial error covariance matrix $[P_o]$ is chosen as a diagonal matrix with

$$p_{ii} = 10^{-6} \qquad i = 1, \ldots, 7$$

The record contains 156 data points that covers 24 hours.

The table below gives the trace of [P] for 156 iterations using the four approaches mentioned previously.

Results

Extended Kalman Filter	Second-order Filter	Least Squares Filter (Invariant Imbedding)	Stochastic approx. using innovations	Number of iterations
6.989 E-6	6.983 E-6	6.987 E-6	6.917 E-6	11
6.972 E-6	6.874 E-6	6.908 E-6	5.932 E-6	31
6.955 E-6	6.789 E-6	6.906 E-6	6.302 E-6	51
6.939 E-6	6.766 E-6	6.909 E-6	6.236 E-6	71
6.923 E-6	6.739 E-6	6.919 E-6	6.133 E-6	91
6.908 E-6	6.536 E-6	6.957 E-6	5.997 E-6	111
6.893 E-6	6.317 E-6	7.080 E-6	5.785 E-6	131
6.879 E-6	6.095 E-6	7.312 E-6	5.352 E-6	151

Table 1: Trace [P]

Methods	CPU Time (sec.) CDC-6400 system
Extended Kalman filter	32.263
Invariant imbedding	84.211
Innovations	374.629
Quasilinearization	422.892

Table 2: Execution Time

CONCLUSION

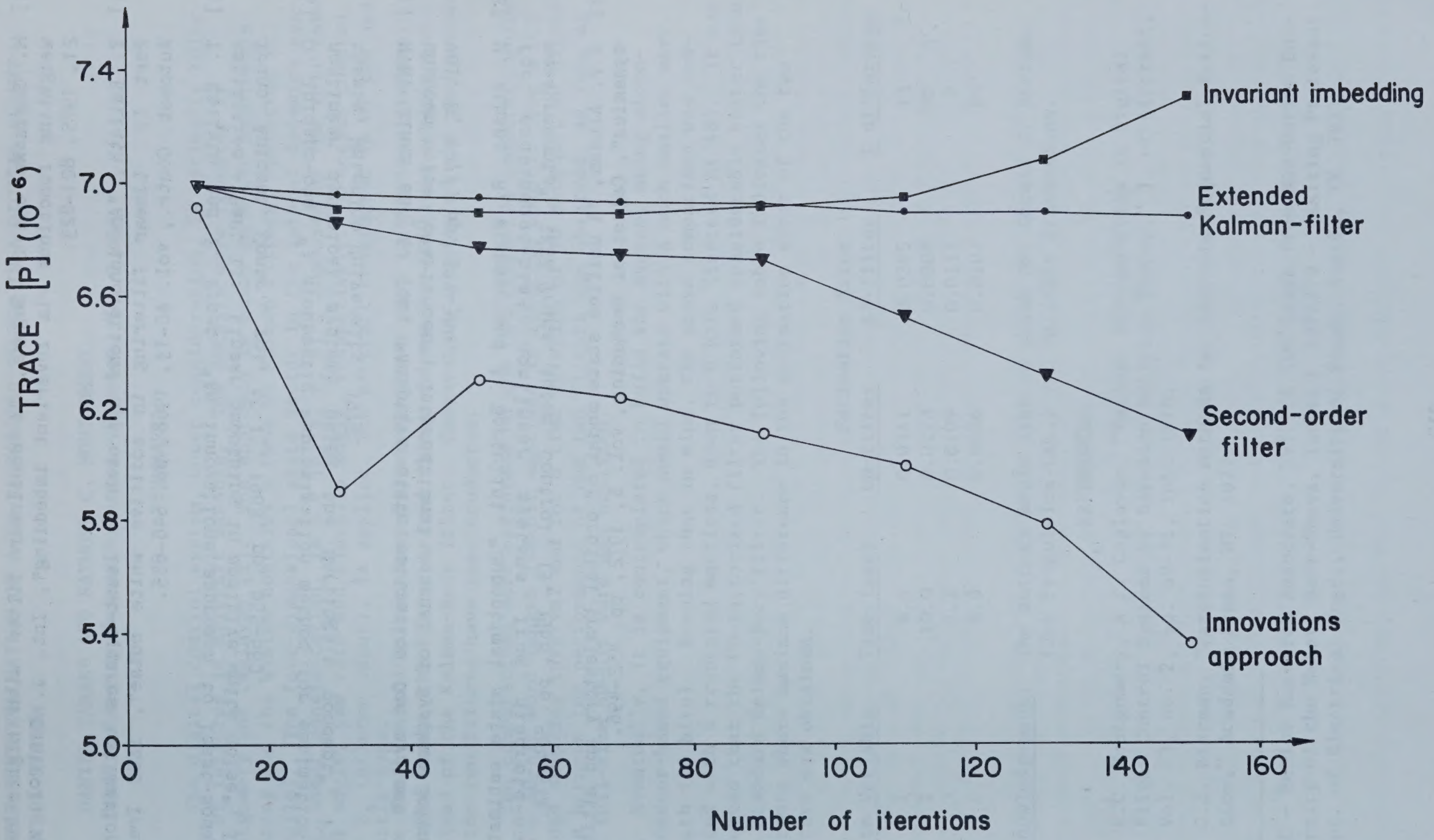
The Innovations approach gives better convergence and takes less time than the Quasilinearization approach, both approaches are second-order filters. The Extended Kalman filter is a first order filter, thus the computations are much less but convergence cannot be assured.

The Invariant Imbedding approach takes much less time compared to other second-order filters, but it diverges in this case.

REFERENCES

[1] N.K. Sinha and S.A. Azim, "Adaptive Estimation of the states of a synchronous-orbit satellite", Int. J. Systems SCI., vol. 9, 1978, pp. 121-1127.

[2] E.S. Lee, "Quasilinearization and Invariant Imbedding", 1968, Academic Press, N.Y.


TRACE [P] (10⁻⁶)
7.4
7.0
6.6
6.2
5.8
5.4
5.0
0
20
40
60
80
100
120
140
160
Number of iterations
Invariant imbedding
Extended Kalman-filter
Second-order filter
Innovations approach

[3] M. Sugisaka and S. Sagara, "A nonlinear filter with higher-order weighting functions via invariant imbedding", Int. J. Control, vol. 21, 1975, 801-823.

[4] T. Kailath, "An innovations approach to least-squares estimation - Part I: Linear filtering in additive white noise", IEEE Trans. Automat. Contr., vol. AC-13, 1968, pp. 646-655.

[5] T. Kailath and P. Frost, "An innovations approach to least-squares estimation - Part II: Linear Smoothing in additive white noise", IEEE Trans. Automat. Contr., vol. AC-13, 1968, pp. 655-660.

[6] O. Ibidapo-obe, "A stochastic approximation method for estimation in nonlinear dynamical systems using the Martingale concept", Ph.D. thesis, Waterloo University, 1976.

[7] N.K. Sinha and A. Tom, "Adaptive state estimation for systems with unknown noise covariances", International Journal of Systems Science, vol. 8, 1977, pp. 377-384.

[8] M. Athans, R. Wishner and A. Bertolini, "Suboptimal state estimation for continuous-time nonlinear systems from discrete-noisy measurements", IEEE Trans. Autom. Control, 13, 1968, pp. 504-514.

[9] S.P. Altman, "A unified state model of orbital trajectory and attitude dynamics", Celestial Mechanics, vol. 6, 1972, pp. 452-496.

COMPARATIVE EVALUATION OF A UD-FACTORIZED FILTER IN A DIGITAL FLIGHT CONTROL SYSTEM

DANIEL TABAK* AND RAYMOND C. MONTGOMERY
NASA - Langley Research Center, Hampton, VA 23665

SUMMARY

The digital control system, under consideration, is optimally tolerant of failures in aircraft sensors. The details were reported by Montgomery and Caglayan [1]. The approach is based on M-ary hypothesis testing and the system contains a bank of Kalman filters, operating in parallel. The four-dimensional state vector in the lateral dynamics system, under consideration, is $x = [p, \phi, r, \beta]^T$, composed of the roll rate, bank angle, yaw rate and sideslip angle, respectively. A scalar control, $u = \delta_a$ (aileron input), is applied. The aircraft considered is of the space shuttle type.

The standard Kalman-Bucy filter has been used in previous studies. In this study, the UD-factorized filter (where $P = UDU^T$) is substituted instead of the Kalman-Bucy filter. Comparative studies of the performance of the two filters have been conducted.

The computations were performed on the CDC CYBER-174 computer, which has a 60-bit word length (48-bit mantissa in floating point). For a 0.1 second sampling interval and for a time duration of 10 seconds, no difference between the performance of the two filters was observed, as long as the full-size 60-bit word was utilized.

Eventually, it is contemplated to build the system, using microprocessor-based equipment, which would operate with a much smaller word length (16 bits). Keeping that in mind, the above comparison was conducted with a truncated mantissa, down to 8 bits (instead of 48). It was observed that the UD-factorized filter performed considerably better than the standard Kalman-Bucy filter. The following table indicates the time instants where maximum difference in the estimation error of the two filters was obtained.

State Variable	Time (sec)	Estimation Error UD-filter	Estimation Error KB-filter	% difference
1	8.4	0.0341	0.0385	13
2	10.0	0.0273	0.0808	200
3	7.2	0.0109	0.0111	2
4	8.9	0.0086	0.0107	24

Acknowledgement: The authors would like to thank Dr. Gerald J. Bierman (Jet Propulsion Lab.) for valuable discussions.

REFERENCES

1. R.C. Montgomery, A.K. Caglayan, "Failure Accommodation in Digital Flight Control Systems by Bayesian Decision Theory," J. of Aircraft, Vol. 13, No. 2, pp. 69-75, Feb. 1976.
2. G.J. Bierman, "Farctorization Methods for Discrete Sequential Estimation," Academic Press, NY, 1977.

---

*NRC - Senior Research Associate, 1977/78 (On leave from Ben-Gurion University of the Negev, Beer-Sheva, Israel.) 1978/79 - Visiting Professor, Dept. of Electrical Engr., University of Texas, Austin, TX 78712.

FLOWSHOP SCHEDULING WITH LIMITED TEMPORARY STORAGE

CHRISTOS H. PAPADIMITRIOU*
Aiken Computation Laboratory
Harvard University
Cambridge, Mass. 02138

PARIS C. KANELLAKIS†
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Mass. 02139

ABSTRACT

We examine the problem of scheduling 2-machine flowshops in order to minimize makespan, using a limited amount of intermediate storage buffers. Although there are efficient algorithms for the extreme cases of zero and infinite buffer capacities, we show that all the intermediate (finite capacity) cases are NP-complete. We prove exact bounds for the relative improvement of execution times when a given buffer capacity is used. We also analyze an efficient heuristic for solving the 1-buffer problem, showing that it has a 3/2 worst-case performance. Furthermore, we show that the "no-wait" (i.e., zero buffer) flowshop scheduling problem with 4 machines is NP-complete. This partly settles a well-known open question, although the 3-machine case is left open here.

1. INTRODUCTION

Flowshop scheduling is a problem that is considered somehow intermediate between single- and multi-processor scheduling [1], [5]. In the version concerning us here, we are given n jobs that have to be executed on a number of machines. Each job has to stay on the first machine for a prespecified amount of time, and then on the second for another fixed amount of time, and so on. For the cases that the (j+1)st machine is busy executing another job when a job is done with the j-th machine, the system is equipped with first-in, first-out (FIFO) buffers, that cannot be bypassed by a job, and that can hold up to b jobs at a time. We seek to minimize the makespan of the job system, in other words, the time between the starting of the first job in the first machine and the end of the last job in the last machine (Let $\mu(S)$ denote the makespan, where S is the schedule).

Some information had been available concerning the complexity of such problems. In the two-machine case, for example, if we assume that there is no bound on the capacity of the buffer ($b = \infty$) we can find the optimum schedule of n jobs in $O(n \log n)$ steps using the algorithm of [6]. Notice that, for $m > 2$, the m-machine, unlimited buffer problem is known to be NP-complete [3]. Also for two machines, when no buffer space is available ($b = 0$, the "no-wait" case) the problem can be considered as a single-state machine problem in the fashion of [2]. In Section 2 of the present paper we show that for any fixed finite size buffer the problem (denoted as (2,b)-FS) becomes NP-complete. In an attempt to construct approximation algorithms we prove in the same section that using 1 buffer can save up to 1/3 of the makespan required without buffer and this generalizes to b

---

*Research supported by NSF Grant MCS77-01192

†Research supported by NSF/RANN grant APR76-12036 and NSF grant DAR78-17826

buffers. We have used this fact to develop a heuristic, which has a 50% worst case behavior for the (2,1)-FS problem but appears to perform much better (4-5% error) on typical problem instances.

In Section 3 we present results that extend our understanding of the complexity of flowshop scheduling under buffer constraints in another direction: we show that the m-machine zero-buffer problem (denoted as (m,0)-FS) is NP-complete for $m \geq 4$. As mentioned earlier, the $m = 2$ case can be solved efficiently. Such "no-wait" problems in general can be viewed as specially structured Traveling Salesman problems, furthermore, it was known that the problem is hard when m is allowed to vary as a parameter [7]. For fixed m and particularly $m = 3$ the complexity of the problem was an open question [5], [7]. Although our proof for $m \geq 4$ is already very complicated, it appears that settling the $m = 3$ case requires a departure from our methodology.

The following exposition presents the theorems and a basic outline of their proofs. Complete proofs and details are contained in [8].

2. COMPLEXITY AND BOUNDS FOR FLOWSHOP SCHEDULING WITH NONZERO BUFFERS

The problem of scheduling 2-machine flowshops in order to minimize makespan, using a fixed, finite, non-zero amount of intermediate buffers is NP-complete. More formally

Theorem 1. For all b, $0<b<\infty$ the (2,b)-FS problem is NP-complete.

The proof is based on a reduction of the following variation of the three way matching of integers, known to be NP-complete [3].

"Given a set A of n integers $A = \{a_1,\dots,a_n\}$ and a set B of 2n integers $B = \{b_1,\dots,b_{2n}\}$ is there a partition of B into n pairs $P_1 = \{p_{i1},p_{i2}\}$ such that for all i $a_i + p_{i1} + p_{i2} = c$ where $c = 1/n(\Sigma a_i + \Sigma b_j)$ (an integer)?"

Let $\mu_b(J)$ be the shortest possible makespan of a job system J using b buffers. In this section we show that

Theorem 2. $\sup\limits_J \dfrac{\mu_0(J)}{\mu_b(J)} = \dfrac{2b+1}{b+1}$.

In other words, the use of b buffers can save up to b/2b+1 of the time needed to execute any job system. Each job i is a pair of numbers (α_i,β_i) denoting execution times on the first and second machines respectively.

Proof. We shall first show that

$$\frac{\mu_0(J)}{\mu_1(J)} \leq \frac{3}{2} \quad .$$

For this purpose, we consider a job system J and an optimal 1-buffer schedule S of length $\mu_1(J)$. We first notice that we can assume that S is a saturated schedule -- that is

$$\mu(S) = \sum_{i=1}^{n} \alpha_i = \sum_{j=1}^{n} \beta_j \ .$$

To see this one just needs to observe that for any $\mathcal{J}$ and S one can create a job system $\mathcal{J}'$ such that $\mu_1(\mathcal{J}') = \mu_1(\mathcal{J})$, $\mu_0(\mathcal{J}') \geq \mu_0(\mathcal{J})$ and S is a saturated schedule for $\mathcal{J}'$, by "filling in" all idle times of S. Given such a saturated schedule S, we create a corresponding schedule S' using no buffers and having the same permutation π of jobs. Let us call a maximal set of consecutive -- in S' -- jobs with no idle time in machine 2 between them a run. We shall construct a 0 buffer schedule for $\mathcal{J}'$ with makespan $\leq 3/2\ \mu_1(\mathcal{J}')$; this will then mean that $\mu_0(\mathcal{J}) \leq \mu_0(\mathcal{J}') \leq 3/2\ \mu_1(\mathcal{J}') = 3/2\ \mu_1(\mathcal{J})$. Our construction will examine each run R separately and will consider two cases.

a) The total idle time s in the first machine during run R is less than or equal to $1/2\ \beta(R)$, where $\beta(R)$ is the total execution time on the second machine of jobs in the run R. In this case our construction leaves R intact.

b) $s > 1/2\ \beta(R)$ (see Figure 1a), and hence R consists of $k+1 \geq 2$ jobs. We first note that

$$s = \sum_{i=1}^{k} (\beta_i - \alpha_{i+1}) > \frac{1}{2}\ \beta(R) \ ,$$

hence

$$\sum_{i=2}^{k+1} \alpha_i + \beta_{k+1} \leq \frac{1}{2}\ \beta(R) \ . \tag{1}$$

We also observe that, in S, the end of α_{k+2} could not have been to the left of the beginning of β_{k+1}, because of the 1-buffer requirement. We conclude that

$$\sum_{i=2}^{k+2} \alpha_i \geq \sum_{j=1}^{k} \beta_k \ . \tag{2}$$

Subtracting (1) from (2) we obtain

$$\alpha_{k+2} \geq \frac{1}{2}\ \beta(R) \ . \tag{3}$$

We thus change π--the optimum permutation of $\mathcal{J}'$ for 1 buffer--by putting J_{k+1} in the end. The corresponding 0-buffer schedule is shown in Figure 1b. The total idle time on the first machine due to the jobs in R is now (see Figure 1b).

$$s' = \sum_{j=1}^{k-1} (\beta_j - \alpha_{j+1}) + \beta_k - \min(\beta_k, \alpha_{k+2}) + \beta_{k+1} \leq \beta(R) - \min(\beta_k, \alpha_{k+2}) .$$

Two cases

1. $\beta_k \geq \alpha_{k+2}$. Then $s' \leq \frac{1}{2} \beta(R)$ by (3).

2. $\beta_k \leq \alpha_{k+2}$. In this case we observe that,

$$s' \leq \sum_{i=1}^{k-1} \beta_1 + \beta_{k+1} \leq \sum_{i=2}^{k+1} \alpha_i + \beta_{k+1} \leq \frac{1}{2} \beta(R) \quad \text{by (1)} \quad .$$

Hence in both cases (a) and (b) our construction succeeds in producing a 0-buffer schedule in which each run R is accountable for machine-1 idle time bounded by 1/2 $\beta(R)$. Hence the total machine-1 idle time is bounded by

$$\frac{1}{2} \sum_{j=1}^{n} \beta_j = \frac{1}{2} \mu_1(J)$$

thus completing the proof of the bound for b=1.

For b<1 although the argument is similar to the one used for the b=1 case examined above, this time it has to be more complicated. As before we first consider the optimum and--without loss of generality--saturated schedule S, with b buffers, for the job system $\mathcal{J}$. We next construct the 0-buffer schedule S' corresponding to the same permutation, which, for simplicity, we take to be the identity. We partition the set of jobs into runs, i.e., maximal sets of jobs without intermediate idle time in the second machine in S', α run with more than one jobs is proper.

For each run R, let f(R) and ℓ(R) be the indices of the first and last jobs of this run, respectively. Also

$$\beta(R) = \sum_{f(R)}^{\ell(R)-1} \beta_j$$

(slightly different from when b=1), γ(R) is the total time during which both machines execute jobs in R in the initial schedule S.

For a run R, let C(R) be the set of indices of jobs subsequent to R that execute concurrently with R, also α_j^R is the portion of J_j that is executed concurrently with R, (in the initial schedule S again).

The proof of the upper bound is based on the following Lemma and an involved accounting scheme used.

Lemma: For each proper run R there exists an index $i(R) \in R$, and $j(R) \in C(R)$ such that $\beta_{i(R)}, \alpha^{R}_{j(R)} \geq \beta(R)-\gamma(R)/b$, $i(R) \neq \ell(R)$.

For a complete proof see [8]. We should also note that the bound $\frac{2b+1}{b+1}$ is achievable in order to justify the equality in Theorem 2.

Consider the following algorithm for obtaining (possibly suboptimal) solutions of the (2,1)-FS problem, for a set of n jobs $\mathcal{J}$.

Algorithm A

1. Solve the 0-buffer problem for $\mathcal{J}$ using the Gilmore-Gomory algorithm [2] to obtain a permutation π of $\mathcal{J}$.
2. Schedule $\mathcal{J}$ with 1 buffer using π.

We must remark that this algorithm can in fact be implemented in O(nlogn), has a worst case error of 50% and an average case error (using simulation) of 4-5%.

3. THE COMPLEXITY OF THE "NO-WAIT PROBLEM

By extending the notation introduced in Section 2 we can define the m-machine no-wait problem as (m,0)-FS. In this section we will prove the following

Theorem 3. The (4,0)-FS problem is NP-complete.

For the purposes of this proof, we introduce next certain special kinds of directed graphs. Let $\mathcal{J}$ be an m-machine job system, and let K be a subset of $\{1,2,\ldots,m\}$. *The digraph associated with $\mathcal{J}$ with respect to K $D(\mathcal{J};K)$* is a directed graph $(\mathcal{J},A(\mathcal{J};K))$, such that $(J_i,J_j) \in A(\mathcal{J};K)$ iff Job J_j can follow job J_i in a schedule S which introduces no idle time in the processors in K.

The definition of the set of arcs $A(\mathcal{J},K)$ given above could be made more formal by listing an explicit set of inequalities and equalities that must hold among the processing times of the two jobs. To illustrate this point, we notice that if m=4 and $K = \{2,3\}$ (Figure 2a) the arc (J_1,J_2) is included in $A(\mathcal{J},K)$ iff we have

$$(1) \quad \alpha_2 \leq \beta_1,\ \gamma_2 \geq \delta_1 \text{ and } \beta_2 = \gamma_1 \quad .$$

We define $\mathcal{D}(m;K)$ to be the class of digraphs D such that there exists a job system $\mathcal{J}$ with $D = D(\mathcal{J};K)$. We also define the following class of computational problems, for fixed $m > 1$ and $K \dot{\subset} \{1,2,\ldots,m\}$

(m,K) Hamilton Circuit Problem

"Given an m-machine job system $\mathcal{J}$, does $D(\mathcal{J};K)$ have a Hamilton circuit?"

We shall prove Theorem 3 by using the following result:

Theorem 4. The (4,{2,3})-Hamilton circuit problem is NP-complete.

We shall prove Theorem 4 by employing a general technique for proving Hamilton path problems to be NP-complete first used by Garey, Johnson and Tarjan [4]. The intuition behind this technique is that the satisfiability problem is reduced to the different Hamilton path problems by creating subgraphs for clauses on one side of the graph and for variables on the other and relating these subgraphs through "exclusive-or gates" and "or gates" (see Figure 3).

First, we have to start with a special case of the satisfiability problem.

Lemma 1. The 3-satisfiability problem remains NP-complete even if each variable is restricted to appear in the formula once or twice unnegated and once negated.

Secondly, realizing special-purpose digraphs in terms of job systems presents us with certain problems. Although our special-purpose digraphs will be similar to those in Figure 4, certain modifications cannot be avoided. A digraph in $\mathcal{D}(4;\{2,3\})$ must be realizable in terms of some job system, so that the inequalities and equations in (1) are satisfied. Care must be taken so that no extra arcs--dangerous to the validity of our argument--are implied in our construction. We shall address this question first.

Consider a digraph $D = (V,A)$, and a node $b \in V$ such that

a) b has indegree and outdegree one.

b) (u,b), $(b,v) \in A$, where outdegree (u) = indegree $(v) = 1$

Then b is called a bond. Removal of all bonds from D divides D into several (weakly connected) components, e.g.,(b_1,b_2,b_3,b_4 in Figure 4a).

Lemma 2. If all components of $D = (V,A)$ are in $\mathcal{D}(4;\{2,3\})$, then $D \in \mathcal{D}(4;\{2,3\})$.

In Figures 5 and 6 we show the components corresponding to each clause (part 1 of Figure 3) and to each variable (2 and 3 of Figure 3) respectively of the Boolean formula we wish to reduce to a graph. We also present realizations of these components via job systems. For details on the proper function of these components when they are connected see [8]. This completes the sketch of the proof of theorem 4. Theorem 3 easily follows from Theorem 4.

Note: One may wish to show that the Hamilton circuit problem is NP-complete for $\mathcal{D}(3;K)$ for some $K \neq \phi$. Now, if $|K| = 2$ the corresponding problem is polynomial. The $|K| = 3$ case and, in general, the Hamilton problems for $\mathcal{D}(m;\{1,2,...m\})$ are equivalent to searching for Euler paths in graphs in which the jobs are represented by arcs and the nodes are the "profiles" of jobs in the Ghannt chart [9]. This leaves us with the interesting open $|K| = 1$ case.

4. CONCLUSIONS

We have investigated the complexity of flowshop scheduling under intermediate storage buffer constraints. The 2-machine finite non-zero buffer case and the fixed $m \geq 4$ machine "no wait" case have been proven NP-complete. Bounds have been derived for the first problem and an approximation algorithm.

Our results of Section 3 leave only the 3-machine case open.

We conclude by examining how much our assumption that the buffer is FIFO affects the resulting scheduling problem. In Figure 2B we show a job system that fares slightly better when the FIFO assumption is removed.

REFERENCES

[1] E.G. Coffman, Jr. (editor), "Computer and Job-Shop Scheduling Theory," Wiley, New York, (1976)

[2] P.C. Gilmore, R.E. Gomory, "Sequencing a One-State Variable Machine: A Solvable Case of the Traveling Salesman Problem," Operations Res., 12 (1964), pp. 655-679.

[3] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, 1978 (to appear).

[4] M.R. Garey, D.S. Johnson, R.E. Tarjan, "The Planar Hamiltonian Circuit Problem is NP-Complete," Siam J. of Computing, 5, 4(1976), pp. 704-714.

[5] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," to appear to Annals of Discrete Math. (1977).

[6] S.M. Johnson, "Optimal Two-and-Three-Stage Production Schedules," Naval Research and Logistics Quarterly 1, 1(1954), pp. 61-68.

[7] J.K. Lenstra, "Sequencing by Enumerative Methods," Mathematisch Centrum, Amsterdam (1976).

[8] C.H. Papadimitriou, P.C. Kanellakis, "Flowshop Scheduling with Limited Temporary Storage", Harvard TR-05-78 or MIT ESL-P-808.

[9] R.L. Rivest, private communication.

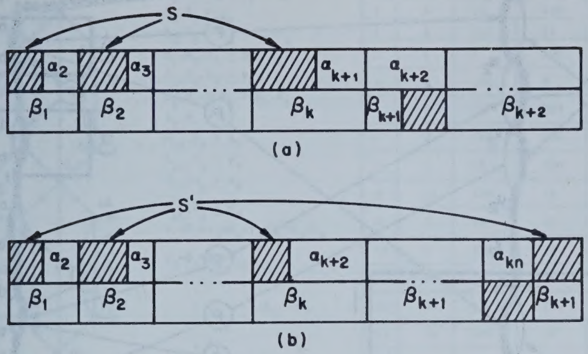

S
α2
α3
αk+1
αk+2
β1
β2
βk
βk+1
βk+2
(a)
S'
α2
α3
αk+2
αkn
β1
β2
βk
βk+1
βk+1
(b)

Figure 1
Modifying one run for the (2,b)-FS bound.

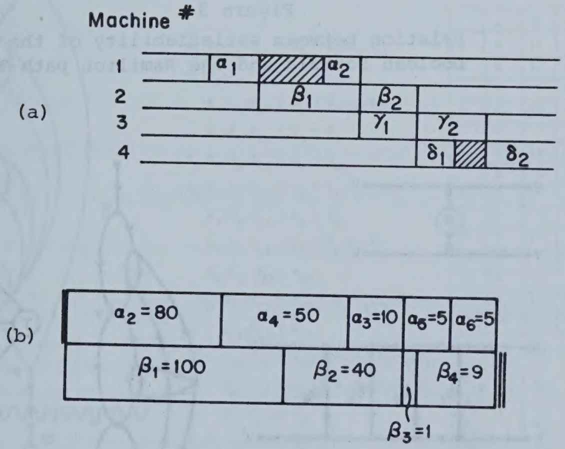
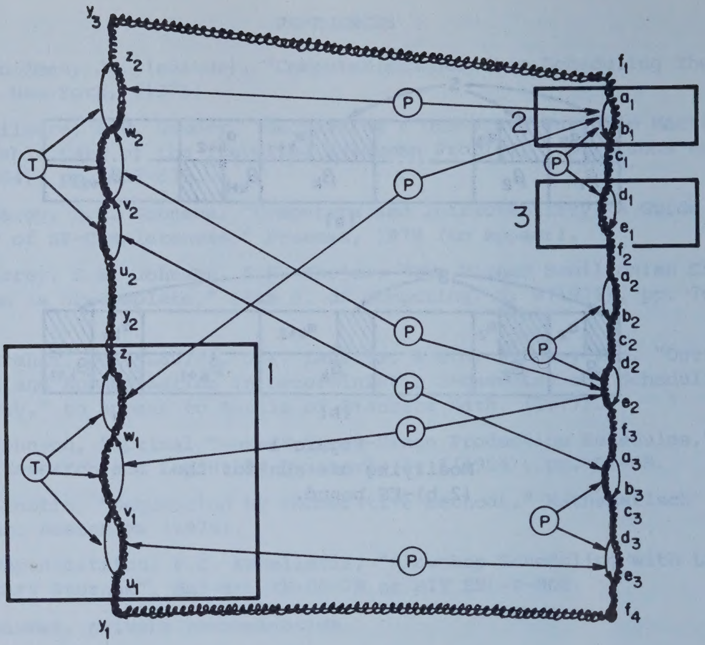

Machine #
1
2
3
4
(a)
α1
α2
β1
β2
γ1
γ2
δ1
δ2
(b)
α2=80
α4=50
α3=10
α5=5
α6=5
β1=100
β2=40
β4=9
β3=1

Figure 2

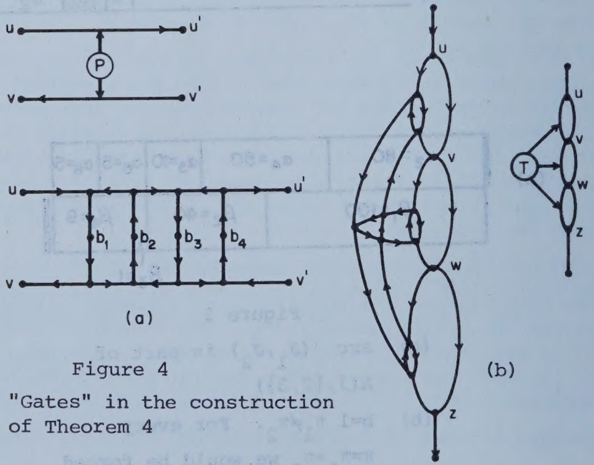
(a) arc (J_1,J_2) is part of $A(J,\{2,3\})$

(b) b=1 $\pi_1 \neq \pi_2$. For every $\pi=\pi_1=\pi_2$ we would be forced to introduce idle time.


y3
z2
w2
v2
u2
y2
z1
w1
v1
u1
y1
T
T
P
P
P
P
P
P
P
P
P
f1
a1
b1
c1
d1
e1
f2
a2
b2
c2
d2
e2
f3
a3
b3
c3
d3
e3
f4
1
2
3

$(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$

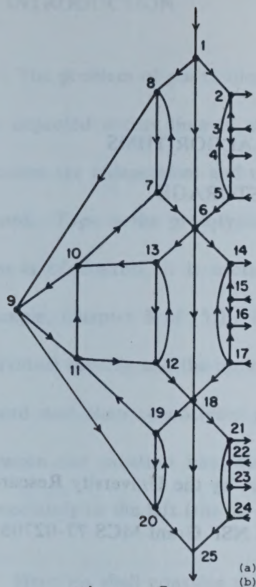
Figure 3
Relation between satisfiability of the boolean formula and the Hamilton path ∿∿∿


u
u'
P
v
v'
u
u'
b1
b2
b3
b4
v
v'
u
v
w
z
T
u
v
w
z

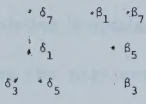
(a)

(b)

Figure 4
"Gates" in the construction of Theorem 4


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

$\beta_1,\beta_3,\beta_5 \geq \alpha_2$
$\delta_1,\delta_3,\delta_5 \leq \gamma_2 \leq \delta_7$
$\beta_3,\beta_5 \geq \alpha_4$
$\delta_3,\delta_5 \leq \gamma_4 < \delta_7,\delta_1$
$\beta_1,\beta_5,\beta_7 \geq \alpha_6 > \beta_3$
$\delta_1,\delta_5,\delta_7 \leq \gamma_6$
$\beta_1,\beta_7 \geq \alpha_8 > \beta_3,\beta_5$
$\delta_1,\delta_7 \leq \gamma_8$
$\beta_{10},\beta_8 \geq \alpha_7$
$\delta_{10},\delta_8 \leq \gamma_7$
$\beta_{10},\beta_8 \geq \alpha_9$
$\delta_{10},\delta_8 \leq \gamma_9$
$\gamma_1=\beta_2=\gamma_3=\beta_4=\gamma_5=\beta_6=\gamma_7=\beta_8$
$\gamma_8=\beta_7=\beta_9=\gamma_{10}$


δ_7
β_1
β_7
δ_1
β_5
δ_3
δ_5
β_3

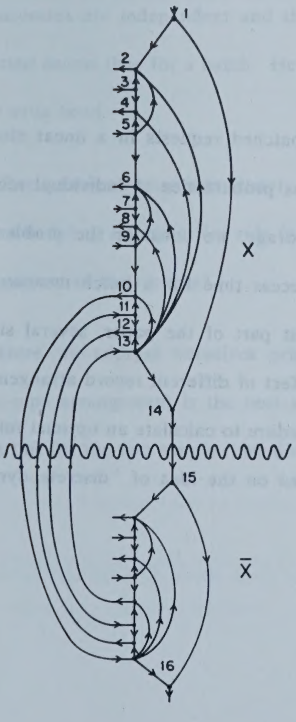
(c)

	α	β	γ	δ
1	0	15	100	10
2	4	100	14	0
3	0	5	100	5
4	4	100	9	0
5	0	10	100	5
6	9	100	200	10
7	0	16	100	15
8	11	100	16	0
9	0	16	17	0
10	0	18	16	0
11	0	17	18	0
12	0	17	200	15
13	11	200	17	0
14	4	200	14	0
15	0	5	200	5
16	4	200	9	0
17	0	10	200	5
18	9	200	300	10
19	11	18	300	0
20	0	300	18	15
21	4	300	14	0
22	0	5	300	5
23	4	300	9	0
24	0	10	300	5
25	9	300	400	0

(b)

(a) component for clause
(b) table of values
(c) inequalities and lattice of β's and δ's for nodes 1 to 10

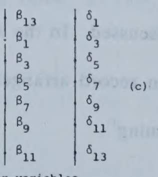
Figure 5


1
2
3
4
5
6
7
8
9
X
10
11
12
13
14
15
$\bar{X}$
16

$\beta_1,\beta_3,\beta_5,\beta_{13} \geq \alpha_2 > \beta_7,\beta_9,\beta_{11}$
$\delta_1,\delta_3,\delta_5,\delta_{13} \leq \gamma_2$
$\beta_3,\beta_5,\beta_{13} \geq \alpha_4 > \beta_7,\beta_9,\beta_{11}$
$\delta_3,\delta_5,\delta_{13} \leq \gamma_4 < \delta_1$
$\beta_5,\beta_7,\beta_9,\beta_{13} \geq \alpha_6 > \beta_{11}$
$\delta_5,\delta_7,\delta_9,\delta_{13} \leq \gamma_6 < \delta_1,\delta_3$
$\beta_7,\beta_9,\beta_{13} \geq \alpha_8 > \beta_{11}$
$\delta_7,\delta_9,\delta_{13} \leq \gamma_8 < \delta_1,\delta_3,\delta_5$
$\beta_9,\beta_{11},\beta_{13} \geq \alpha_{10}$
$\delta_9,\delta_{11},\delta_{13} \leq \gamma_{10} < \delta_1,\delta_3,\delta_7,\delta_5$
$\beta_{11},\beta_{13} \geq \alpha_{12}$
$\delta_{11},\delta_{13} \leq \gamma_{12} < \delta_3,\delta_5,\delta_7,\delta_9,\delta_1$
$\beta_1 , \beta_{13} \geq \alpha_{14} > \beta_3,\beta_5,\beta_7,\beta_9,\beta_{11}$
$\delta_1 , \delta_{13} \leq \gamma_{14}$
$\gamma_1=\beta_2=\gamma_3=\beta_4=\gamma_5=\beta_6=\gamma_7=$
$=\beta_8=\gamma_9=\beta_{10}=\gamma_{11}=\beta_{12}=\gamma_{13}=\beta_{14}$

	α	β	γ	δ
1	0	11	15	13
2	6	15	14	0
3	0	9	15	11
4	6	15	12	0
5	0	7	15	9
6	2	15	10	0
7	0	5	15	7
8	2	15	8	0
9	0	3	15	5
10	0	15	6	0
11	0	1	15	3
12	0	15	4	0
13	0	13	15	1
14	10	15	14	0

(b)


β_{13}
β_1
β_3
β_5
β_7
β_9
β_{11}
δ_1
δ_3
δ_5
δ_7
δ_9
δ_{11}
δ_{13}
(c)

(a) components for variables
(b) table of values
(c) inequalities and lattice of β's and δ's

Figure 6

OPTIMAL AND NEAR-OPTIMAL SCHEDULING ALGORITHMS FOR BATCHED PROCESSING IN LINEAR STORAGE

J. R. Bitner[1] and C. K. Wong[2]

1. Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

The research of this author was supported in part by the University Research Institute at the University of Texas at Austin and NSF Grant MCS 77-02705.

2. IBM T. J. Watson Research Center
P. O. Box 218
Yorktown Heights, N. Y. 10598

Abstract. In this paper, we consider the accessing of batched requests in a linear storage medium. The batch size is assumed fixed and the access probabilities of individual records known. For a given arrangement of records in the storage, we consider the problem of read/write head scheduling to minimize the expected access time for a batch measured in terms of the distance traveled by the head. In the first part of the paper, several simple algorithms are proposed, analyzed and compared. The effect of different record arrangements is also discussed. In the second part of the paper, a procedure to calculate an optimal rule for any given record arrangement is descibed, which is based on the idea of "discrete dynamic programming".

1. INTRODUCTION

The problem of positioning a set of records in a linear storage medium in such a way that the expected access time is minimized has been thoroughly studied [1-4] when consecutive accesses are independent and the frequencies are all known prior to the placement of the first record. Tape is the prototypical linear storage medium but, when minimization of disk seek time is of interest, it is useful to view the cylinders as forming a linear store. (See, for example, Chapter 5 of [5]). It is sufficient to know the relative frequency of access to the individual records and the optimal solution is obtained by placing the most frequently accessed record and then repetitively placing the next most frequently accesses record alternating between the position immediately to the right of those already placed and the position immediately to the left (the so-called organ-pipe arrangement).

Here we shall consider the accessing of batched requests, i.e. we process a fixed number (a batch) of b requests at a time. We make the same assumption as before, namely, consecutive accesses are independent and the frequencies are known. Our objective is to minimize expected access time for a batch. Here access time is measured by the distance traveled by the read/write head.

Two problems arise immediately. First, for a given arrangement of the records, what is a good scheduling algorithm (or rule for short) for the head movement? Second, what is a good arrangement of the records?

Here we address ourselves primarily to the first problem since we believe that the organ-pipe arrangement is the best arrangement for most reasonable rules. In fact, we can prove this for the two simple rules discussed later.

2. FORMULATION OF THE PROBLEM

Let the locations of a linear storage be labeled 1,2,...,n from left to right. We use a row n-vector to represent the arrangement of records, e.g. $(R_1,...,R_n)$ means record R_i is at location i for all i. Let p_i be the access probability of record R_i for $i = 1,2,...,n$. Then $\sum_{i=1}^{n} p_i = 1$. At time t, b requests are generated (with repetitions allowed) according to these probabilities. Let $L_1 \leq L_2 \cdots \leq L_b$, $1 \leq L_i \leq n$, be the b locations where the requests are made. From now on, L_1, L_b will be referred to as the left and right extremes respectively and will be written as L and R. Suppose the current head location is x, $1 \leq x \leq n$. A rule specifies the order in which the head goes through $L_1, L_2, ..., L_b$, starting from x. The head stops when all locations have been visited. Let the stopping location be y. y will be the starting location for the next batch, i.e. for time $t + 1$. Let d(x,y) denote the distance traveled by the head from x to y, whose expected value is our cost. Here we are only interested in rules whose cost has a definite value. For example, rules whose behavior can be modeled by a Markov chain. A more formal definition of rules will be given later. Our objective is to find rules with as small a cost as possible.

3. NEAR-OPTIMAL ALGORITHMS

In this section, we propose two simple rules which are called the Leftist and Alternating rules.

Leftist rule. From the current head location x, move to the left extreme L, then sweep across to the right extreme R, and stop.

A Rightist rule can be similarly defined. That is, the head moves to the right first and then sweep left.

Alternating rule. When the time t is odd, use Leftist rule, otherwise use Rightist rule.

Let $COST_L$, $COST_R$, $COST_A$ be the cost functions of the Leftisit, Rightist and Alternating rules respectively. Also, define

$$\lambda_i = \text{Prob}(L \geq i) = \left(\sum_{j=i}^{n} p_i\right)^b$$

$$\rho_i = \text{Prob}(R \leq i) = \left(\sum_{j=1}^{i} p_i\right)^b$$

By symmetry, $\text{COST}_L = \text{COST}_R$. It can be shown that

$$\text{COST}_L = 2 \sum_{i=1}^{n-1} (1-\lambda_{i+1})(1-\rho_i)$$

$$\text{COST}_A = \sum_{i=1}^{n-1} (1-\rho_i^2-\lambda_{i+1}^2)$$

and that COST_L (COST_R), COST_A are all minimized when the record arrangement is the organ-pipe arrangement.

Based on these results, it can be further shown that

(1) $\text{COST}_A \leq \text{COST}_L = \text{COST}_R$

for any record arrangement.

(2) $C_{min} \leq C_A \leq 2 \cdot C_{min}$,

where C_{min} is the minimum cost among all rules and all record arrangements; C_A is COST_A when the organ-pipe arrangement is used.

Therefore the Alternating rule is better than the Leftist or Rightist rule and is never more than twice larger than the minimum.

A slightly more complicated but intuitively appealing rule is the following.

Nearest rule. From the current head location, move to the closer of the two extremes, then sweep across to the other extreme and stop. In case of tie, choose either extreme.

Analysis of this rule is harder than for previous rules because it is difficult to calculate the probability that the head will be at a given position after accessing a batch.

The head position can be described by a Markov chain where the i-th state ($i=1,\ldots,n$) corresponds to having the head at location i. It is not difficult to write down the transition matrix $[p_{xy}]$, where p_{xy} = Prob (head ends up at y | head starts at x). But solving for the steady state distribution $(s_1,\ldots,s_n)$ symbolically is difficult and $\mathrm{COST}_N = \sum_{i=1}^{n-1} (1 - \lambda_{i+1} - \rho_i) + \sum_{i=1}^{n} \sum_{u \le r} s_i \cdot \mathrm{Prob}(L = u, R = r) \cdot \min(|i-u|, |i-r|)$.

Numerical computations were carried out for several standard record access probability distributions. For all cases, $\mathrm{COST}_N \le \mathrm{COST}_A$. However, one can construct a distribution such that this is not true. Also, it is not known if the organ-pipe arrangement minimizes COST_N.

4. OPTIMAL ALGORITHM

All the previous rules are simple to execute, but their costs are not optimal. Next, we present a procedure to calculate an *optimal* rule for a given record arrangement. The amount of computation depends on the arrangement and the access probabilities. We follow the idea of "discret dynamic programming" developed by Blackwell [11].

The situation in discrete dynamic programming is as follows: we are given a system with T states (labeled $1,\ldots,T$). At any time, the system is in exactly one state. At intervals of time, we are required to choose any one of a given set of A *actions* (labeled $1,\ldots,A$) to be performed. The *cost* of performing action a in state t is given by $c(t,a)$. (We will assume all costs are non-negative.) After an action is performed, the system moves to a new state. The probability that action a will cause the system to move from state t to t' is denoted by $p(t \rightarrow t' \mid a)$. A *decision function*, f is a function from $\{1,\ldots,T\}$ into $\{1,\ldots,A\}$ where $f(t)$ specifies that action to be performed in state t. A *policy* $\pi = (f_1, f_2, \ldots,)$ is a sequence of decision functions where f_i is used to determine the i-th decision.

The correspondence between this terminology and the original problem should be clear: a state corresponds to an ordered triple (x, ℓ, r) where x is the current head position and ℓ and r are the extremes of the current batch. There are only two actions: one for moving to the left extreme first, then sweeping across to the right and another for moving right first. (Note that when we apply an action in a given state, only the x of the next state is determined; ℓ and r are chosen probabilistically.) The correspondence for costs and probabilities follow directly.

It can be shown that for our case an optimal rule is a uniform rule, i.e. a rule consisting of one decision function only, which is used to access all batches. This is intuitively clear since the batches are independent and each batch will have an infinite number following, there is no intuitive justification for using different decision functions for accessing two different batches.

The procedure for calculating an optimal rule starts with an arbitrary rule (we choose the Nearest rule) and iteratively improves it. At each iteration we assume the current rule will be used to access all batches except the first and then calculate the best rule for accessing the first batch. It can be shown that the asymptotic cost for this rule is less than or equal to that for the original rule. This new rule is then used on the next iteration, and we continue iterating until the new rule and the current rule are identical. The procedure for calculating an-optimal rule is given below.

1. Make the Nearest rule the "current" rule.

2. Calculate $f_B(x)$ (the cost of accessing B batches with the head starting at position x) for the "current" rule. $f_B(x)$ is iteratively calculated for $B = 1,2,\ldots$ using the formula

$$f_0(x) = 0$$

$$f_B(x) = \sum_{\ell,r} \mathrm{Prob}(L = \ell, R = r) \bullet \delta(x,\ell,r)$$

where

$$\delta(x,\ell,r) = \begin{cases} |x - \ell| + (r - \ell) + f_{B-1}(r) & \text{if the current rule goes to } \ell \text{ first with the head at x and extremes } \ell \text{ and r} \\ |x - r| + (r - \ell) + f_{B-1}(\ell) & \text{if it goes to r first.} \end{cases}$$

The calculation continues until we reach a B such that $f_B(x) - f_{B-1}(x)$ is nearly constant with respect to x.

3. Calculate the "new" rule as follows:

For each x, ℓ and r, if

$$|x - r| - |x - \ell| + \lim_{B\to\infty} [f_B(\ell) - f_B(r)] < 0$$

the "new" rule should move to the right extreme first with head at x and extremes ℓ and r. If this quantity is non-negative, the rule should move left first. (Note that $\lim_{B\to\infty} [f_B(\ell) - f_B(r)]$ is closely approximated by $f_B(\ell) - f_B(r)$ for some large B.)

4. If the new rule is the same as the current rule, stop. This rule is the ∞-optimal rule. Otherwise set the "current rule" to be the "new rule" and go to step 2.

REFERENCES

1. P. P. Bergmans, Minimizing expected travel time on geometrical patterns by optimal probability rearrangements, Information and Control, 20 (1972), pp. 331-350.
2. D. D. Grossman and H. F. Silverman, Placement of records on a secondary storage device to minimize access time, J. Assoc. Comput. Mach., 20 (1973), pp. 429-438.
3. V. R. Pratt, An N log N algorithm to distribute N records optimally in a sequential access file, Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds. Plenum Press, New York, 1972, pp. 111-118.
4. P. C. Yue and C. K. Wong, On the optimality of the probability ranking scheme in storage applications, J. Assoc. Comput. Mach., 20(1973), pp. 624-633.
5. E. G. Coffman, Jr. and P. J. Denning, Operating Systems Theory, Prentice-Hall, New Jersey, 1973.
6. R. M. Karp, A. C. McKellar and C. K. Wong, Near-optimal solutions to a 2-dimensional placement problem, SIAM J. Comput., Vol. 4, No. 3 (1975), pp. 271-286.
7. P. C. Yue and C. K. Wong, Near-optimal heuristics for an assignment problem in mass storage, Int. J. Comput. and Inf. Sc., Vol. 4, No. 4, (1975), pp. 281-294.
8. A. C. McKellar and C. K. Wong, Dynamic Placement of Records in Linear Storage, IBM Research Report RC 6410, 1977 (also to appear - JACM).
9. B. Shneiderman and V. Goodman, Batched Searching of Sequential and Tree Structured Files, ACM Trans. on Database Systems, Vol. 1, No. 3, September 1976, pp. 268-275.
10. P. B. Hoel, S. C. Port and C. J. Stone, Introduction to Stochastic Processes, Houghton Mifflin Company, Boston, 1972.
11. D. Blackwell, Discrete Dynamic Programming, Annals of Mathematical Statistics, 33 (1962), pp. 719-726.

DETERMINISTIC SCHEDULING WITH PIPELINED PROCESSORS

JOHN BRUNO, JOHN JONES and KIMMING SO
Department of Electrical Engineering and Computer Science and the
Computer Systems Laboratory, University of California,
Santa Barbara, CA 93106

ABSTRACT

In this paper we prove several results showing a correspondence between problems involving task systems with single-stage processors and similar ones using pipelined processors. For example, an optimal schedule for a task system with arbitrary precedence constraints using a single pipelined processor with two stages is found by the familiar Coffman-Graham algorithm for a system with two single-stage processors. Precedence constraints in the form of in-forests, out-forests, and directed acyclic graphs are examined. Task systems with release-times and deadlines for each task are also considered.

1. INTRODUCTION

In this paper we consider the problem of scheduling a finite collection of tasks on one or more pipelined processors. Our results are divided into three major categories depending on the type of precedence constraints. Each problem we consider is similar to a problem with one or more single-stage processors and in each case we give the reference for the corresponding problem.

In our model a processor consists of a number of _stages_ and the processing of a task corresponds to having the task visit each stage in a fixed sequence. For example, let P be a processor consisting of s stages and let T be a task. Assigning task T to stage 1 _initiates_ processing of T by P. After one unit of time T moves on to stage 2 for additional processing and so on until T finally reaches stage s. When T leaves stage s we say that P _completes_ processing task T. The important feature is that when T moves to stage 2 a new task, say T', may be initiated by P. When T moves to stage 3 and T' to stage 2, P may again initiate the processing of a new task. Continuing in this way we see that it is possible for P to be processing s tasks simultaneously. Throughout this paper we assume that each task requires one unit of processing time at each of the s stages of a processor. Consequently, the processing time for a task assigned to a processor (the time from initiation on the first stage to completion on the last stage) is equal to the number of processor stages, namely s.

An instance of a scheduling problem consists of a finite collection of tasks $\mathcal{T} = \{T_1,\ldots,T_n\}$, precedence constraints given by a partial ordering $<$ on the members of $\mathcal{T}$, a deadline d_i and a release-time r_i for each task T_i, and a number m of processors, each one consisting of s stages. The partial ordering $<$ will usually be defined by giving directed acyclic graph G (with vertex set $\mathcal{T}$) and letting $<$ correspond to the transitive closure of G. This partial ordering represents certain technological constraints which are enforced by requiring that a task may not initiate processing until all of its predecessors have completed their processing. The pair $(\mathcal{T}, <)$ is called a _task system_.

---

* This work was partially supported by National Science Foundation grant MCS75-22557.

Example: Let $\mathcal{T} = \{T_1,\ldots,T_{12}\}$ and $\prec$ be given by the directed acyclic graph in Figure 1(a). Release-times and deadlines are shown in Figure 1(b). In Figure 1(c) we show the progress of each task as it moves from stage-to-stage within a processor (s=3). For example, task T_7 is initiated at time 8 on processor P1 with task T_6 at stage 3 and an idle stage 2. We are constrained not to initiate a task prior to its release-time and that is why task T_6 is not initiated until time 6 even though the immediate predecessor of 6, T_2, completed at time 4.

Not every task is completed on or before its deadline. For example, $d_8 = 5$ and T_8 completes at time 11 making the lateness of task T_8 equal to 6. This is the maximum lateness for the given schedule. The makespan of this schedule is 14.

It is redundant to show the progress of each task as it moves from stage-to-stage within a processor and from now on we will only show the initiation time of each task (see Figure 1(d)).

In the next section we present our results. For proofs, see [BJS].

2. RESULTS

We consider three classes of problems depending on the nature of the precedence constraints: namely, in-forests, out-forests and arbitrary precedence constraints. In the first two cases we allow $m \geq 1$ identical processors with $s \geq 1$ stages and in the last case a single processor with two stages.

In each case we give an algorithm for constructing a list $L = (T_{i_1}, T_{i_2}, \ldots, T_{i_n})$ of the tasks in $\mathcal{T}$ such that list scheduling[†] based on L yields a schedule with the desired property.

Let $\{g_i\}$ be a set of labels associated with $\mathcal{T} = \{T_1,\ldots,T_n\}$ where g_i corresponds to T_i for i=1,...,n. Let $L = (T_{i_1},\ldots,T_{i_n})$ be an ordered list of the members of $\mathcal{T}$. We say that L is decreasing with respect to $\{g_i\}$ if $g_{i_j} \geq g_{i_{j+1}}$ for j=1,...,n-1. We say that L is increasing with respect to $\{g_i\}$ if $g_{i_j} \leq g_{i_{j+1}}$ for j=1,...,n-1. All of our results are summarized in Table 1.

2.1 In-Forests

We say that $\prec$ is an in-forest if each task in $\mathcal{T}$ has at most one immediate successor. The partial ordering of the task system in Figure 2 is an in-forest. Associated with each task T_i in $\mathcal{T}$ is a deadline d_i and a release-time r_i. In this subsection $r_i = 1$ for i=1,...,n ($r \equiv 1$). Our objective is to construct a schedule which minimizes the maximum lateness, i.e., minimizes $\max_i\{\sigma(T_i) + s - d_i\}$ over all schedules σ, where $\sigma(T_i)$ is the initiation

---

[†] List scheduling based on L assigns a free processor to the leftmost ready task in L.

tiation time of T_i under schedule σ and s is the number of processor stages. Our results follow closely those of Brucker et al. [BGJ] for m identical single stage processors.

Table 1: Summary of Results

Problem Instance					Objective	Results		Reference for single-stage version
$\prec$	m	s	$\{r_i\}$	$\{d_i\}$		Theorem	List†	
In-Forest	≥ 1	≥ 1	$r \equiv 1$	arbitrary	minimize maximum lateness	1	$\uparrow\{d_i'\}$	BGJ
In-Forest	≥ 1	≥ 1	$r \equiv 1$	not applicable	minimize makespan	2	$\downarrow\{\lambda_i\}$	Hu
Out-Forest	≥ 1	≥ 1	$r \equiv 1$	not applicable	minimize makespan	3	$\downarrow\{\delta_i\}$	DL
Arbitrary	1	2	positive integers	positive integers	feasibility	4	$\uparrow\{d_i\}$	GJ1
Arbitrary	1	2	$r \equiv 1$	not applicable	minimize makespan	5	$\downarrow\{\alpha_i\}$	CG

†All schedules are list schedules.

$\uparrow\{g_i\} \equiv$ increasing list with respect to $\{g_i\}$

$\downarrow\{g_i\} \equiv$ decreasing list with respect to $\{g_i\}$

The construction of L is based on the idea of modified deadlines. Let $\{d_i\}$ be the original deadlines. The modified deadlines $\{d_i'\}$ are computed using the following algorithm.

Algorithm MD (Modified Deadline)

1. If T_i is a task with no successors, then assign $d_i' \leftarrow d_i$.
2. If T_i is a task which has not been assigned a modified deadline and whose immediate successor T_j has been assigned a modified deadline, then assign $d_i' \leftarrow \min(d_i, d_j' - s)$ where s is the number of processor stages. □

Theorem 1. Let $(\mathcal{T}, \prec)$ be a task system where $\prec$ is an in-forest, $m \geq 1$ and $s \geq 1$. Let $\{d_i\}$ be the original deadlines, $r \equiv 1$, $\{d_i'\}$ the modified deadlines computed using Algorithm MD, and L an increasing list with respect to $\{d_i'\}$. Then the schedule obtained by applying list scheduling using L minimizes the maximum lateness.

In Figure 2 we give an example of a modified deadline computation, construction of L and a list schedule based on L.

Using Theorem 1 and the computation scheme for modified deadlines we can show that the level schedules of T. C. Hu [Hu] minimizes the makespan. We compute the level λ_i of a task T_i as follows.

Algorithm LEVEL

```
1. If $T_i$ is a task with no successors then assign $\lambda_i \leftarrow 1$.
2. If $T_i$ is a task which has not been assigned a level and whose
   immediate successor $T_j$ has been assigned a level, then assign
   $\lambda_i \leftarrow \lambda_j + 1$.
```

Let $(\mathcal{T}, \prec)$ be a task system and $\prec$ be an in-forest. Suppose $d_i = d$ for $i=1,\ldots,n$. If we compare the modified deadline computation MD with the level computation LEVEL, it is not difficult to see that if L is an increasing list with respect to $\{d_i'\}$, then L is a decreasing list with respect to $\{\lambda_i\}$. Moreover, if we take $d = 1$ then maximum lateness corresponds to makespan. Thus we have:

Theorem 2. Let $(\mathcal{T}, \prec)$ be a task system where $\prec$ is an in-forest, $r \equiv 1$, $m \geq 1$ and $s \geq 1$. Let $\{\lambda_i\}$ be computed using LEVEL and L a decreasing list with respect to $\{\lambda_i\}$. Then a schedule obtained by applying list scheduling using L minimizes the makespan.

2.2 Out-Forests

We say that $\prec$ is an out-forest if each task in $\mathcal{T}$ has at most one immediate predecessor. The partial ordering of the task system depicted in Figure 3 is an out-forest. It is shown in [BGJ] that when $\prec$ is an out-forest then the problem of determining whether a schedule exists which meets all the task deadlines is NP-complete [Ka]. Since that is just the special case $s = 1$ we turn our attention elsewhere.

If our intention is to minimize the makespan then there is a well-known way of relating out-forests to in-forests. Namely, if we reverse all the relations in the out-forest $\prec$ we obtain the in-forest $\prec_R$. If we construct a schedule σ_R which minimizes makespan for $(\mathcal{T}, \prec_R)$ and we run the schedule σ_R in reverse, then we obtain a minimum makespan schedule for $(\mathcal{T}, \prec)$. This construction works for all $m \geq 1$ and $s \geq 1$.

However, there is a labeling scheme which works directly on out-forests for the case $s = 1$ [DL] which also minimizes makespan for $s > 1$.

First we give a scheme for computing a δ value for each task in $\mathcal{T}$.

Algorithm CP (Critical Path)

```
1. If $T_i$ has no immediate successor then set $\delta_i \leftarrow 1$.
2. If $T_i$ has not been assigned a $\delta$ value and all the immediate suc-
   cessors of $T_i$, namely, $T_{j_1},\ldots,T_{j_k}$, have $\delta$ values, then assign
   $\delta_i \leftarrow \max\{\delta_{j_1},\ldots,\delta_{j_k}\} + 1$.
```

Theorem 3. Let $(\mathcal{T}, \prec)$ be a task system where $\prec$ is out-forest, $r \equiv 1$, $m \geq 1$ and $s \geq 1$. Let $\{\delta_i\}$ be computed using CP and L a decreasing list with respect to $\{\delta_i\}$. Then a schedule obtained by applying list scheduling using L minimizes the makespan.

An example is given in Figure 3. The proof of Theorem 3, given in [BJS], provides a detailed analysis of the structure of these minimum makespan schedules.

2.3 Arbitrary Precedence Constraints, $m=1$, $s=2$

In this section we extend known results for $m=2$ and $s=1$ [GJ1] to the case $m=1$ and $s=2$. It is interesting to note the correspondence between the number of single stage processors (2) on the one hand and the number of stages (2) in the lone processor on the other.

We follow [GJ1] in our development. Let $(\mathcal{T},\prec)$ be a task system, $\{r_i\}$ a set of positive integer release-times and $\{d_i\}$ a set of positive integer deadlines. Our goal is to construct a feasible schedule† for $(\mathcal{T},\prec)$ on one processor $(m=1)$ with two stages $(s=2)$.

For any task T_i and integers r, d satisfying $r_i \leq r \leq d_i \leq d$, we define S(i,r,d) to be the set of all tasks T_j $(j \neq i)$ which have $d_j \leq d$ and either have $r \leq r_j$ or $T_i \prec T_j$. Let N(i,r,d) denote the number of tasks in S(i,r,d).

We can now state and prove an important lemma which justifies certain deadline modifications.

Lemma 1: For any task T_i and integers r, d satisfying $r_i \leq r \leq d_i \leq d$ and for any feasible schedule we have:

i) If $N(i,r,d) = d-r-1$, then T_i must be completed by $d-N(i,r,d)$

ii) If $N(i,r,d) > d-r-1$, then T_i must be completed by $d-N(i,r,d)-1$.

Proof:

i) Suppose $N(i,r,d) = d-r-1$ and let σ be any feasible schedule. Since all the tasks in S(i,r,d) have deadlines which do not exceed d and $s=2$, all tasks in S(i,r,d) must start before d-1. If all the tasks in S(i,r,d) start no sooner than r then T_i must start by time $r-1 = d-N(i,r,d)$. Otherwise some task T_j in S(i,r,d) starts before r. By the definition of S(i,r,d), $T_i \prec T_j$ and thus T_i must be completed before r.

ii) Suppose $N(i,r,d) > d-r-1$. Then at least $N(i,r,d)-d+r+1$ tasks in S(i,r,d) start before r. By the definition of S(i,r,d), T_i precedes each of these tasks and thus T_i is completed by time $r-(N(i,r,d)-d+r+1) = d-N(i,r,d)-1$.

As a consequence of this lemma, if $N(i,r,d) = d-r-1$ and $d_i > d-N(i,r,d)$ then we can set d_i to $d-N(i,r,d)$, or if $N(i,r,d) > d-r-1$ and $d_i > d-N(i,r,d)-1$, then we can set d_i to $d-N(i,r,d)-1$; and these changes will not alter the set of feasible schedules. Following [GJ1] we say that $\{d_i\}$ and $\{r_i\}$ are internally consistent whenever the following conditions hold for every task $T_i \in \mathcal{T}$:

---

† A feasible schedule is one in which no task completes after its corresponding deadline or initiates before its release-time.

1. $d_i \geq r_i + 2$
2. For every pair of integers r, d satisfying $r_i \leq r \leq d_i \leq d$
   - i) If $N(i,r,d) = d-r-1$, then $d_i \leq d-N(i,r,d)$
   - ii) If $N(i,r,d) > d-r-1$, then $d_i \leq d-N(i,r,d)-1$.

Theorem 4. Let $(\mathcal{T},\prec)$ be a task system with $m = 1$, $s = 2$. Let $\{d_i\}$ and $\{r_i\}$ be internally consistent and L an increasing list with respect to $\{d_i\}$. Then the schedule obtained by applying list scheduling with respect to L is feasible.

Our scheduling algorithm consists of repeatedly modifying deadlines using Lemma 1 until either we obtain a d_i satisfying $d_i < r_i + 2$ or $\{d_i\}$ and $\{r_i\}$ are internally consistent. In the former case there is no feasible schedule and in the latter case Theorem 4 shows us how to construct a feasible schedule. See [GJ1] for an $O(n^3)$ algorithm for obtaining an internally consistent pair $\{r_i\}$ and $\{d_i\}$.

In order to minimize the makespan when $r \equiv 1$ we can assign $d \equiv w$ and search for the smallest w which yields a feasible schedule. There is an alternative based on the labeling algorithm of [CG].

Let $N = (n_1,\ldots,n_t)$ and $N' = (n'_1,\ldots,n'_{t'})$ be two decreasing sequences of positive integers, i.e., $n_i > n_{i+1}$ and $n'_j > n'_{j+1}$ for $1 \leq i < t$ and $1 \leq j \leq t'$. We say that $N < N'$ if either

1. For some i, $1 \leq i \leq t$, we have $n_j = n'_j$ for $1 \leq j < i$ and $n_i < n'_i$, or
2. $t < t'$ and $n_j = n'_j$ for $1 \leq j \leq t$.

Algorithm CG (Coffman-Graham)

Initially all tasks are unlabeled, i.e., α_i is undefined. An unlabeled task is said to be ready if all of its successors are labeled. If T_i is a ready task then N_i denotes the decreasing sequence of α-labels of the successors of T_i.

1. set $j \leftarrow 1$.
2. Let R be the set of ready tasks. If $R = \phi$. stop
3. Let $T_i \in R$ be such that $N_i \leq N_j$ for all $T_j \in R$.
4. set $\alpha_i \leftarrow j$, $j \leftarrow j+1$ and go to step 2.

Theorem 5. Let $(\mathcal{T},\prec)$ be a task system with $m = 1$ and $s = 2$. Let $\{\alpha_i\}$ be computed using the CG algorithm and L a decreasing list with respect to $\{\alpha_i\}$. Then a schedule obtained by applying list scheduling using L minimizes the makespan.

REFERENCES

[GJ1] Garey, M. R. and Johnson, D. S., "Two-Processor Scheduling with Start-Times and Deadlines," SIAM J. Computing, v6, n3, Sept. 1977, pp. 416-426.

[CG] Coffman, E. G., Jr. and Graham, R. L., "Optimal Scheduling for Two-Processor Systems," Acta. Informatica, v1, 1972, pp. 200-213.

[Ka] Karp, R. M., "Reducibility among Combinatorial Problems," Complexity of Computer Computations, R. E. Miller and J. M. Thatcher, eds., Plenum Press, New York, 1972, pp. 45-68.

[DL] Davida, G. I. and Linton, D. J., "A New Algorithm for Scheduling Tree Structured Tasks," Proceedings of the 1976 Conference on Information Sciences and Systems, Department of Electrical Engineering, Johns Hopkins University, pp. 543-548.

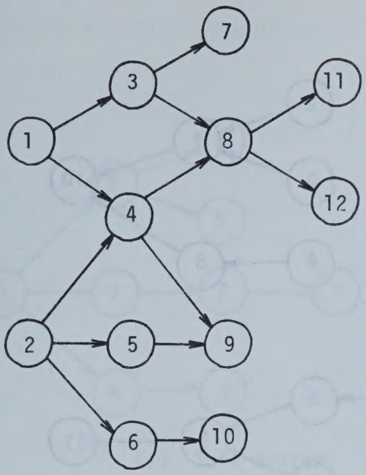
[BGJ] Brucker, P., Garey, M. R. and Johnson, D. S., "Scheduling Equal-Length Tasks under Treelike Precedence Constraints to Minimize Maximum Lateness," Mathematics of Operations Research, v2, n3, 1977, pp. 275-284.

[GJ2] Garey, M. R. and Johnson, D. S., "Scheduling Tasks with Non-uniform Deadlines on Two Processors," JACM, v23, n3, July 1976, pp. 461-467.

[Hu] Hu, T. C., "Parallel Sequencing and Assembly Line Problems," Operations Research, 9 (1961), pp. 841-848.

[Co] Computer and Job Shop Scheduling Theory, E. G. Coffman, Jr. (ed), John Wiley and Sons, 1976.

[BJS] Bruno, J., Jones, J. and So, K., "Deterministic Scheduling with Pipelined Processors," TR-CSL-7802, Computer Systems Laboratory, University of California, Santa Barbara, CA.



(a) Task System

i	1	2	3	4	5	6	7	8	9	10	11	12
d_i	8	7	6	5	12	15	17	5	9	16	14	14
r_i	1	1	5	2	3	6	2	6	1	10	5	5

(b) Deadlines and Release Times

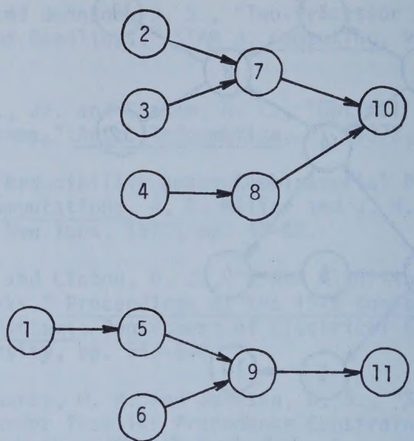
P_2	stage 3			1			4	3		9	8			12	
	stage 2		1			4	3		9	8			12		
	stage 1	1			4	3		9	8			12			time
		1			4			7			10			13	14
P_1	stage 3			2			5		6		7		10	11	
	stage 2		2			5		6		7		10	11		
	stage 1	2			5		6		7		10	11			time
		1			4			7			10			13	14

(c) A Schedule

P2	1	-	-	4	3	-	9	8	-	-	12	-	-
P1	2	-	-	5	-	6	-	7	-	10	11	-	-
	1			4			7			10			13 14

(d) Initiation Schedule

Figure 1. A Task System and A Schedule



(a) Task System (in-forest)

d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}
8	9	5	6	2	10	10	7	12	7	9

(b) Deadlines

d'_1	d'_2	d'_3	d'_4	d'_5	d'_6	d'_7	d'_8	d'_9	d'_{10}	d'_{11}
0	3	3	3	2	5	5	5	7	7	9

(c) Modified Deadlines (s = 2)

$(T_1, T_5, T_2, T_3, T_4, T_6 . T_7, T_8, T_9, T_{10}, T_{11})$

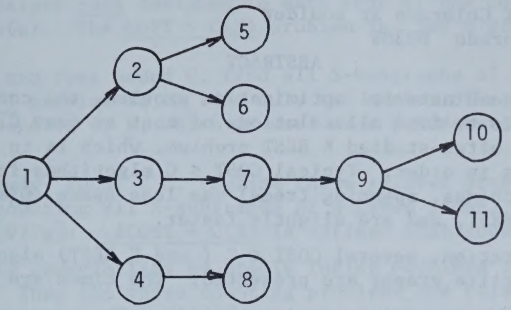
(d) Increasing List with Respect to $\{d'_i\}$

P2	2	4	6	8	-	-	-	-
P1	1	3	5	7	9	10	11	-

1 2 3 4 5 6 7 8 9

(e) List Schedule (maximum lateness = 3)

Figure 2. In-Forest



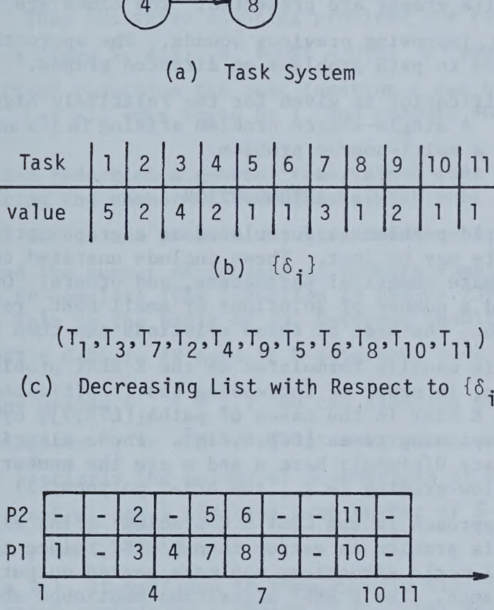
(a) Task System

Task	1	2	3	4	5	6	7	8	9	10	11
δ value	5	2	4	2	1	1	3	1	2	1	1

(b) $\{\delta_i\}$

$(T_1, T_3, T_7, T_2, T_4, T_9, T_5, T_6, T_8, T_{10}, T_{11})$

(c) Decreasing List with Respect to $\{\delta_i\}$



P2	-	-	2	-	5	6	-	-	11	-
P1	1	-	3	4	7	8	9	-	10	-

(d) List Schedule Based on L with s = 2
(makespan = 11)

Figure 3. Out-Forest

FINDING THE BEST SOLUTIONS: THE COST ≤ C PROBLEM AND FINDING MATCHINGS

HAROLD N. GABOW
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309

ABSTRACT

For a given combinatorial optimization problem, the corresponding COST ≤ C problem is to find all solutions of cost at most C. This is closely related to the often-studied K BEST problem, which is to find the K smallest solutions in order. Typical COST ≤ C algorithms for graph problems (e.g., paths, matchings, spanning trees) use less space ($O(m+n)$ instead of $O(K+m+n)$) than K BEST, and are slightly faster.

As an illustration, several COST ≤ C (and K BEST) algorithms for matchings on bipartite graphs are presented. The times are $O(K \min(n^3, mn \log n))$, improving previous bounds. The approach is to reduce the matching problems to path problems on directed graphs.

Lastly, a justification is given for the relatively high time bound on COST ≤ C for paths: A single-source problem arising in its solution is shown to be equivalent to a multi-source problem.

1. INTRODUCTION

When a real-world problem is formulated as a graph optimization problem, various aspects may be lost. These include unstated constraints or desiderata, approximate numerical parameters, and others. One way to deal with this is to find a number of solutions of small cost, rather than just one optimum solution. The best of these solutions can then be selected.

This approach is usually formulated as the K BEST problem: find the K smallest solutions, in order of increasing cost. Efficient algorithms have been given for K BEST in the cases of paths [L72,Y], cycles [JK], matchings [M], and spanning trees [CFM,G,KIM]. These algorithms use time $O(K \cdot P(m,n))$ and space $O(K+m+n)$; here n and m are the number of vertices and edges, and P is a slow-growing (e.g., low order polynomial) function.

An alternate approach is the COST ≤ C problem: find all solutions of cost at most C. This problem is easier than K BEST, since the output need not be sorted. Real-world situations may make sorted output unnecessary or of limited significance. The K BEST algorithms mentioned above can be adapted to COST ≤ C. The space bounds reduce to $O(m+n)$. The constant of proportionality in the time bounds decrease. So for reasons of applicability and efficiency, COST ≤ C may be a better problem formulation than K BEST.

Section 2 gives general remarks on COST ≤ C. These include #P-completeness results [V77a-b], and the space savings mentioned above. Section 3 studies a specific example, COST ≤ C for matchings on bipartite graphs. Algorithms are obtained by reducing the matching problems to path problems. A K BEST matching algorithm improves previous results. Section 4 studies the complexity of COST ≤ C for paths. A possible justification for the relatively high time bound for paths is given, by showing a multi-source path problem is involved.

2. THE COST ≤ C PROBLEM: GENERAL REMARKS

This Section discusses these general complexity properties of COST ≤ C: Counting COST ≤ C solutions is usually #P-complete; a COST ≤ C algorithm usually requires less space than the corresponding K BEST algorithm.

We start with some terminology. Let S-subgraph denote a fixed family of subgraphs of graphs, e.g., cycle, spanning tree, etc. In a graph G, let c(e) be a real-valued cost assigned to each edge e; the cost of a subgraph H is $c(H) = \sum_{e \in H} c(e)$. The $COST \le C(S)$ problem is, for a given graph G with cost function c and cost bound C, find all S-subgraphs of cost at most C. The $\#COST \le C(S)$ problem is to find only the number of such subgraphs. This number is denoted by K, which is used as a parameter in time estimates for $COST \le C$.

We first study $\#COST \le C$. Define these related problems: $\#(S)$ is the problem of counting all S-subgraphs of a given graph. $\#(S)$ problems are studied in [V77b]. $\#COST = C(S)$ is defined analogous to $\#COST \le C$.

Let α denote reducibility by oracle Turing machines in the sense of Valiant [V77b]. Then the three counting problems are related as follows:

Lemma 2.1: For a fixed S, $\#(S) \ \alpha \ \#COST = C(S) \ \alpha \ \#COST \le C(S)$.

Proof: In the first reduction the cost function c can be restricted to positive integers: Let all edge costs be 1, and compute $\#COST = i$, for $i = 0, \ldots, m$.

In the second reduction a general real-valued cost function can be allowed, by computing the number of S-subgraphs with cost $\le C$ and $\ge C$. □

Let #SUBSET SUM be the problem of, given numbers $s_1, \ldots, s_n$ and a desired sum S, find the number of subsets of numbers summing exactly to S. This problem is #P-complete (The reduction in [MY,pp.239-243] leads to a simple proof). This is the basis of our hardness result.

Lemma 2.2: $\#COST \le C(S)$ is #P-hard if S-subgraphs have these two properties:

(a) For any graphs G_i, $i = 1,2$, we can find (in polynomial time) a graph G containing the G_i as edge-disjoint subgraphs, such that the S-subgraphs of G are precisely the unions of S-subgraphs of G_i;

(b) Some graph G_0 has a positive even number of S-subgraphs, with some edge e_0 in exactly half of them.

Proof: By Lemma 2.1, it suffices to show $\#COST = C(S)$ is #P-hard. Use (a) to construct a graph containing n copies of G_0. Let all edge costs be 0, except the i^{th} copy of e_0 has cost s_i. Take $C = S$. Then if G_0 has 2k S-subgraphs, the answer to #SUBSET SUM is the answer to $\#COST = C$ divided by k^n. □

This reduction uses cost 0 edges. However, by scaling up the numbers s_i, we see $COST \le C$ is #P-hard when costs are all positive integers.

$\#(S)$, and hence $\#COST \le C(S)$, is known to be #P-complete when an S-subgraph is a maximum matching or a path joining designated start and end vertices [V77b]. However the Lemma gives interesting results in these cases:

Theorem 2.1: $\#COST \le C(S)$ is #P-complete when an S-subgraph is a maximal matching or spanning tree.

Proof: For a maximal matching, G_0 is a cycle of length 4. For a spanning tree, G_0 is a cycle of length 4 plus a diagonal; the diagonal is in 4 of the 8 spanning trees. □

The status of #(*S*) is unknown for maximal matchings [V77b]. For spanning trees #(*S*) can be solved in polynomial time [E].

Now we turn our attention to COST $\le$ C algorithms. In spite of the previous results, these algorithms give feasible computations if K is not huge. The usual approach is backtracking: edges are systematically included and excluded until all desired *S*-subgraphs are found.

As an example, suppose in a directed graph, an *S*-subgraph is an s-t path, i.e., a simple path from a designated start vertex s to an end vertex t. K BEST (s-t path) has been solved in $O(K \min(n^3, mn \log n))$ time and $O(K+m+n)$ space (see [L72,L76,Y] for time, [G] for space). This approach gives a COST $\le$ C algorithm using the same time but less space, $O(m+n)$.

The algorithm is given below. P is a global data structure for a path. The recursive procedure EXTEND (v) is called with P as the last s-t path found so far, and v on P. Let P = (s,...,v,w,...,t). Then EXTEND (v) outputs all paths (including P) of cost $\le$ C that begin with (s,...,v). This is done by first (recursively) outputting the paths that include edge (v,w), and then the paths that exclude it. EXTEND (v) returns with P set to (s,...,v).

```
procedure PATHS; comment finds all s-t paths of cost ≤ C, where vertices
s,t and cost C have already been specified; begin
procedure EXTEND (v);
begin let EXCLUDE be a list of edges local to EXTEND;
1.  if v = t then begin output (P); return end;
2.  let w be the vertex following v in P;
3.  EXTEND (w); comment now P ends at w;
4.  remove w from P; let EXCLUDE consist of the edge (v,w);
5.  while there is a v-t path avoiding vertices in P-v and edges in EXCLUDE,
          with cost ≤ C-c(P) do
          begin
6.        let $(v=w_0,w_1,\ldots,w_k=t)$ be such a path;
7.        for i ← 1 to k do add $w_i$ to P;
8.        EXTEND ($w_1$);
9.        remove $w_1$ from P; add $(v,w_1)$ to EXCLUDE;
10. end   end EXTEND;
11. let d(v) be the length of a shortest s-v path, for each vertex v;
12. if d(t) > C then return comment no paths of desired cost exist;
13. if G has negative edges then
14.      begin for each edge (v,w) do c(v,w) ← c(v,w) + d(v) - d(w);
15.            C ← C-d(t);
         end;
16. let P be a shortest s-t path;
17. EXTEND (s);
    end PATHS;
```

Lemma 2.3 PATHS solves COST $\leq$ C(s-t path) in a directed graph with no negative cycles, in time $O(K \min(n^3, m\, n \log n))$ and space $O(m+n)$.

Proof: The time is dominated by lines 5-6: finding an acceptable v-t path. This is a single-source shortest path problem with non-negative edge costs (by lines 13-15). So Dijkstra's algorithm does lines 5-6 once in time $O(\min(n^2, m \log n))$[AHU]. These lines are done $\leq n$ times per output path. ☐

This result illustrates the general space savings for COST $\leq$ C.

Theorem 2.2: COST $\leq$ C (S) can be solved in $O(m+n)$ space when an S-subgraph is an s-t path, a cycle, a maximum matching, or a spanning tree in an undirected or directed graph.

Proof: As with s-t paths, other K BEST algorithms [JK,M,G,KIM,CFM] can be modified for backtracking. (Section 3 gives more details for cycles and matchings). ☐

Regarding time, corresponding COST $\leq$ C and K BEST algorithms have the same asymptotic time bound. But a simple comparison of code shows COST $\leq$ C saves time, by replacing a priority queue with a stack.

3. FINDING BEST MATCHINGS

This Section presents algorithms for several COST $\leq$ C problems for matchings on bipartite graphs. Two previous algorithms are improved or extended. The approach is to reduce the matching problems to path problems on directed graphs.

We first recall some fundamental facts. A matching M is a set of disjoint edges. A vertex not on an edge of M is exposed. A path is alternating if its edges are alternately in M and $\overline{M}$, and further, an end edge is in $\overline{M}$ only if the corresponding end vertex is exposed. (This restriction is not standard.) An alternating path is augmenting if both end edges are in G -M, and de-augmenting if these edges are in M. The cost of an alternating path P with respect to M is $c(P,M) = c(P \cap \overline{M}) - c(P \cap M)$.

The following "direct sum principle" is well-known [L76]: Fix a matching M of cardinality k; let N be another cardinality k-matching. Then $N \oplus M$ consists of disjoint alternating cycles, even length alternating paths, and pairs of disjoint augmenting and de-augmenting paths ("Alternating" means alternating with respect to M). Further $c(N \oplus M, M) = c(N) - c(M)$.

The direct sum principle gives a 1-1 correspondence between cardinality k matchings and sets of alternating paths described above. Note when k gives a perfect matching, the sets contain only alternating cycles; when k gives a maximum matching, the sets contain only alternating cycles and even paths.

Now we recall a well-known correspondence between bipartite graphs G with a distinguished perfect matching M and directed graphs D. Specifically, let G have vertices u_i, v_i, i=1, ..., n, and let $M = \{(u_i,v_i) \mid i=1,\ldots,n\}$. The corresponding directed graph D has vertices w_i, i=1, ..., n, and edges (w_i,w_j), where (u_i,v_j) is an edge of G-M.

Alternating cycles in G correspond to directed cycles in D. If edge costs satisfy $c(w_i,w_j) = c(u_i,v_j) - c(u_j,v_j)$, then the costs of an alternating cycle (with respect to M) and the corresponding directed cycle are equal.

Thus by the direct sum principle, a perfect matching N on G corresponds to a set of disjoint cycles S on D, and $c(N) - c(M) = c(S)$. This implies the following result.

Lemma 3.1: Let $F(n,m)$ be the time to find a minimum cost perfect matching on a bipartite graph, $M(n,m,K)$ be the time for COST ≤ C (perfect matching) on a bipartite graph, and $S(n,m,K)$ be the time for COST ≤ C (set of disjoint cycles) on a directed graph with no negative cycles. Then

$$M(n,m,K) = O(F(n,m) + Kn + S(n,m,K)),$$
$$S(n,m,K) = O(Kn + M(n,m,K)).$$

The space for both COST ≤ C problems is equal.

Proof: Given G, find a minimum cost perfect matching M. The corresponding graph D has no negative cycles, by the minimality of M. The equation for M follows. □

Thus COST ≤ C (perfect matching) reduces to COST ≤ C (set of disjoint cycles). This in turn reduces to COST ≤ C (cycle), by the following general technique.

For a fixed $\mathcal{S}$-subgraph, let an $\mathcal{S}$-set be a set of disjoint $\mathcal{S}$-subgraphs. Let FIND (C) be an algorithm for COST ≤ C ($\mathcal{S}$). Often we can convert FIND to an algorithm for COST ≤ C ($\mathcal{S}$-set), by these changes: First, introduce a list LIST to store the $\mathcal{S}$-set. LIST is initially empty. For each call output (H) when an $\mathcal{S}$-subgraph H is found, substitute these lines:

```
1.  begin add H to LIST; remove the vertices of H from the graph;
2.  output (LIST);
3.  FIND (C-c(H));
4.  remove H from LIST; add the vertices of H to the graph; end;
```

Lemma 3.2: Let FIND, an algorithm for COST ≤ C ($\mathcal{S}$), satisfy this property:

(*) Let H_1 and H_2 be disjoint $\mathcal{S}$-subgraphs, with H_1 output before H_2. When H_2 is output, FIND has modified the graph so H_1 is not an $\mathcal{S}$-subgraph.

Then the above modifications to FIND give an algorithm for COST ≤ C ($\mathcal{S}$-set) having the same time and space bounds as FIND.

Proof: Use induction on the number of $\mathcal{S}$-subgraphs. Property (*) insures that an $\mathcal{S}$-set is listed only once. □

Note (*) often holds. For instance if FIND works by finding min cost $\mathcal{S}$-subgraphs, H_1 must be removed before H_2 can be discovered.

We use these results to get algorithms for two versions of COST ≤ C. First consider finding all perfect matchings (i.e., all costs are 0). Lemmas 3.1-2 reduce this to finding all directed cycles. The most efficient algorithms for all cycles use time $O(Km+m+n)$ and space $O(m+n)$ [J75,RT,SL]. This gives the following result.

Theorem 3.1: All perfect matchings on a bipartite graph can be found in $O(n^{1/2}m+n+Km)$ time and $O(n+m)$ space.

Proof: The first perfect matching is found in time $O(n^{1/2}m+n)$ [HK]. □

This result has been previously obtained by Itai et al [IRT]. Our algorithm is essentially equivalent.

Next consider arbitrary cost functions. COST ≤ C (cycle) on a directed graph with no negative cycles can be solvedin $O(K\min(n^3, mn\log n))$ time and $O(m+n)$ space [JK]: The algorithm just repeatedly chooses a vertex s, uses

PATHS of Section 2 to find all s-s paths of cost $\le C$, and then deletes s.

Theorem 3.2: COST $\le C$ (perfect matching) on a bipartite graph can be solved in $O(K\min(n^3, mn\log n))$ time and $O(m+n)$ space.

Proof: A min cost perfect matching M is found in time $O(\min(n^3, mn\log n))$ [L76]. The directed graph corresponding to G,M has no negative cycles by the minimality of M. ☐

A corresponding K BEST algorithm, using the same time and $O(K+m+n)$ space, is easily constructed. This improves the $O(Kn^4)$ time algorithm given in [M].

Next we consider COST $\le C$ (maximum matching). First we specify how a graph with an arbitrary matching corresponds to a directed graph. Let G be bipartite, with vertex set (U,V), and arbitrary matching M. The corresponding directed graph is found by enlarging G to a graph G' with a perfect matching, finding the corresponding directed graph D', and enlarging D' to the desired graph D.

To form G', add a vertex x' for each exposed vertex x of G, and add the matched edge (x,x'). G' has a perfect matching, so let D' correspond to G'. Note the vertices of D' correspond naturally to the matched edges of G', and so partition into 3 sets:

W_M (vertices of D' corresponding to matched edges of M);

W_U (vertices corresponding to matched edges (x,x'), $x \in U$)

W_V (vertices corresponding to matched edges (x,x'), $x \in V$).

To form D, add a fourth set of vertices W_M', containing a vertex w' for each $w \in W_M$, and add the edge (w',w) for each corresponding pair w',w.

As with perfect matchings, alternating cycles of G correspond to directed cycles of D. An even length alternating path starting at an exposed vertex of U corresponds to a directed path from W_U to W_M. An even alternating path starting in V corresponds to a directed path from W_M' to W_V.

Let the edges (w',w) from W_M' to W_M have cost $-c(u,v)$, where w corresponds to $(u,v) \in M$. Then the correspondence between cycles and paths preserves cost.

This correspondence shows COST $\le C$ (maximum matchings) reduces to COST $\le C$ (set of disjoint cycles and S_i-T_i paths, i=1,...,k). (An S-T path, for sets of vertices S, T is a path from some $s \in S$ to some $t \in T$). We allow a number of corresponding sets S_i, T_i). It is easy to see COST $\le C$ (S-T path) reduces to COST $\le C$ (s-t path). Thus the directed graph problem reduces to COST $\le C$ (set of disjoint cycles and s_i-t_i paths, i=1, ..., k). This problem is solved using the following general technique.

Let an $S_1,\ldots,S_k$-set be a set of disjoint S_1-subgraphs, ..., and S_k-subgraphs. If FIND i(C) is an algorithm for COST $\le C$ (S_i-set), i=1, ..., k, then an algorithm for COST $\le C(S_1,\ldots,S_k$-set) can usually be formed as follows: Let FIND1 be the main routine. Assume all FINDi use LIST to store the S_i-set. In FINDi, i=2, ..., k, remove the initialization of LIST. In FINDi, i=1, ..., k-1, add a call, FINDi+1 (C), on entry; also replace each occurence of output (LIST) by these lines:

1. begin output (LIST);
2. remove the vertices in subgraphs of LIST from the graph;
3. FINDi+1 (C-c(LIST));
4. add the vertices in subgraphs of LIST to the graph; end;

Lemma 3.3: Let FINDi be an algorithm for COST ≤ C(S_i-set), i=1, ..., k. Suppose no S_i-subgraph has negative cost, for i=2, ..., k. Then the above modifications to FINDi give an algorithm for COST ≤ C($S_1,\ldots,S_k$-set), whose time and space bounds are the maximum of those for FINDi, i=1, ..., k. □

Now we extend Theorems 3.1-2 to maximum matchings. When costs are 0, all s-t paths can be found using essentially the same algorithm as for cycles [RT].

Theorem 3.3: All maximum matchings on a bipartite graph can be found in time $O(n^{1/2}m+n+Km)$ time and $O(m+n)$ space. □

An equivalent algorithm is given by Itai et al [IRT].
For arbitrary costs, we use PATHS of Section 2 for s-t paths.

Theorem 3.4: COST ≤ C (maximum matching) on a bipartite graph can be solved in $O(K\min(n^3, mn\log n))$ time and $O(m+n)$ space. □

Finally we consider COST ≤ C (cardinality k matching). Let G be a bipartite graph with a matching M of cardinality k; let D be the corresponding directed graph defined above. Augmenting and de-augmenting paths in G corresponds to W_U-W_V and $W_{M'}$-W_M paths in D, respectively. Thus COST ≤ C (cardinality k matching) reduces to COST ≤ C (set of disjoint cycles, S_1-T_1 paths, S_2-T_2 paths, and S_1-T_2, S_2-T_1 path pairs). (An S_1-T_2, S_2-T_1 path pair consists of an S_1-T_2 path and an S_2-T_1 path, disjoint from each other.) Vertex sets S_1, T_1, S_2, T_2 are disjoint; they correspond to W_U, W_M, $W_{M'}$, W_V, respectively.

To solve the COST ≤ C problem on D, we use a graph D*. Form D* by adding two vertices z_i, i=1,2. Add cost 0 edges (w,z_i), where $w \in T_i$, and (z_i,w), where $w \in S_i$, i=1,2. Note the cycles in D* correspond in D to cycles, S_1-T_1 paths, S_2-T_2 paths, and S_1-T_2, S_2-T_1 path pairs.

Unfortunately, the problem in D does not reduce to all cycle-sets in D*. In the direct sum principle, a cardinality k matching corresponds to a set containing an equal number of augmenting and de-augmenting paths, i.e., these paths are not naturally paired. This means in D, sets of path pairs composed of the same paths are not distinct. We will generate sets of S_1-T_2, S_2-T_1 path pairs so (numbers of) the start vertices of both paths increase in successive pairs. This way, a set of path pairs is generated only once.

Another misfortune is that to generate all path pairs, there seems to be no way to avoid generating all cycles of D*. Although this involves extra work, we shall see it is not excessive.

The following algorithm, executed on D*, solves COST ≤ C (S_1-T_2,S_2-T_1 path pair-set) for D.

```
procedure PAIRS (C);
 1. begin for each cycle of cost ≤ C do
 2.           if the cycle corresponds to an $S_1$-$T_2$, $S_2$-$T_1$ path pair P then
 3.           begin add P to LIST;
 4.           remove from D* the vertices of P;
 5.           let $s_i$ be the start vertices of P, $s_i$, $\varepsilon\ S_i$, $i=1,2$;
 6.           remove from D* all edges $(z_i, s)$, for $s \varepsilon S_i$ with $s < s_i$, $i=1,2$;
 7.           output (LIST);
 8.           PAIRS (C-c(P));
 9.           add the vertices and edges removed in lines 4 and 6 to D*;
10.           remove P from LIST;
    end       end PAIRS;
```

Lemma 3.4: PAIRS solves COST ≤ C (S_1-S_2,S_2-T_1 path pair-set). If COST ≤ C (cycle) can be solved in time O(K • T(n,m)), and K_{D*} is the number of cycle sets of cost ≤ C in D*, then PAIRS uses time O(K_{D*} • T(n,m)) and space O(m+n).

Proof: Each cycle found in line 1 corresponds to a distinct cycle-set of D*. □

To solve the original COST ≤ C problem on D, we use PAIRS (on D*) to find all desired pair sets. For each pair set, we find all sets of disjoint cycles, S_1-T_1 paths, and S_2-T_2 paths, using previous algorithms (on D). Note if K is the number of solutions to the entire COST ≤ C problem, then in Lemma 3.4, $K_{D*} \le K$. Thus the pair sets are found in acceptable time.

Theorem 3.4: All cardinality k matchings on a bipartite graph can be found in $O(n^{1/2}m+n+Km)$ time and O(m+n) space. □

Theorem 3.5: COST ≤ C (cardinality k matching) on a bipartite graph can be solved in $O(K\min(n^3, mn\log n))$ time and O(m+n) space. □

These two results are the most general of this Section. Theorem 3.4 extends the results of [IRT]. Theorem 3.5 is easily modified for K BEST (cardinality k matching); the time remains the same and the space increases to O(K+m+n). This extends and improves the $O(Kn^4)$ time algorithm for K BEST (perfect matching) of [M].

4. THE SECOND SHORTEST PATH

The COST ≤ C (s-t path) problem is essentially equivalent to finding a second shortest s-t path. (A COST ≤ C algorithm can be constructed from a second path algorithm, using the approach of PATHS of Section 2; the time per path for COST ≤ C is the time for the second path.) This Section shows finding a second shortest path is equivalent to a variant of the all pairs shortest path problem. The best algorithms for the latter use $O(\min(n^3, mn\log n))$ time [J77]. So this Section gives some justification for the seemingly high time bound for PATHS of Section 2.

We start by formally defining two problems on directed graphs with no negative cycles.

SP: Given vertices s, t, and a value L, does the second shortest s-t path have length $\le L$?

AP: Given values $\ell(v,w)$ for each pair of distinct vertices v, w, does some v-w path have length $\le \ell(v,w)$?

Let S(n) and A(n) denote the time to solve SP and AP, respectively, on an n vertex graph. (Note the restriction in AP that v and w are distinct is inessential. Allowing bounds $\ell(v,v)$ on non-trivial cycles does not change the time complexity A.)

Now we show S(n) and A(n) are essentially equal. First note the structure of a second shortest s-t path:

Lemma 4.1: If P is a shortest s-t path, there are vertices v and w on P, with v preceding w, and a v-w path Q disjoint from P-v-w, such that a second shortest s-t path consists of Q plus the s-v and w-t portions of P. (Possibly s=v or w=t). ☐

Lemma 4.2: $S(n) = O(n^2+A(n))$.

Proof: Let an instance of SP be given on a directed graph G. Let P be a shortest s-t path; if v precedes w on P, let p(v,w) be the length of the portion of P from v to w. Let M be the largest magnitude of an edge length. Define an instance of AP on a graph G', as follows. G' is G with the edges of P deleted. The bound $\ell(v,w)$ is L-p(s,v) - p(w,t), if v precedes w on P; otherwise it is -nM. It is easy to see the instances of SP and AP have the same answer. ☐

Lemma 4.3: $A(n) = O(n^2+S(n))$.

Proof: Let an instance of AP be given on a graph G, having vertices v_i, i=1, ..., n. Without loss of generality, assume G has no negative edges [J77]. Let M be greater than the largest magnitude of a path length bound $\ell(v_i,v_j)$.

Define an instance of SP on a graph G' with edge lengths given by c, and length bound L = nM, as follows. G' contains a copy of G. The additional vertices are u_i, i=1, ..., n+1, where $s = u_1$, $t = u_{n+1}$. The additional edges are in a shortest path from s to t: (u_i,u_{i+1}), i=1, ..., n, with $c(u_i,u_{i+1}) = 0$; also edges leaving the path (u_i,v_i), i=1, ..., n, with $c(u_i,v_i) = (n-i)M$; and edges entering the path: (v_j,u_{i+1}), i,j=1,..., n with $c(v_j,u_{i+1}) = iM-\ell(v_i,v_j)$.

Note all edge lengths are non-negative. Thus a shortest s-t path is $(u_1,\ldots,u_{n+1})$. Now we show the instances of AP and SP have the same answer.

If AP has answer "yes" then some v_i-v_j path has length $\le \ell(v_i,v_j)$. Adding on the paths $(u_1,\ldots,u_i,v_i)$ and $(v_j,u_{i+1},\ldots,u_{n+1})$ gives an s-t path of length $\le$ nM. So SP has answer "yes".

Conversely, suppose SP has answer "yes". A second shortest s-t path consists of a path in G from v_i to v_j, plus paths $(u_1,\ldots,u_i,v_i)$ and $(v_j,u_{k+1},u_{k+2},\ldots,u_{n+1})$, where $k \ge i$. Its length is $(n-i+k)M - \ell(v_k,v_j) + p$, where p is the length of the v_i-v_j path. For this to be $\le$ nM, we must have k = i, and $p \le \ell(v_i,v_j)$. Thus AP has answer "yes". ☐

Combining Lemmas 4.2-3 gives the desired result.

Theorem 4.1 The time for SP is $O(n^2)$ if and only if the time for AP is $O(n^2)$. Otherwise, the time for SP is the same order as the time for AP. □

ACKNOWLEDGMENT

The author thanks Frank Fussenegger of Martin Marietta Coroporation for his inspiring discussions on matching.

REFERENCES

[AHU] A. Aho, J. Hopcroft and J. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[CFM] P. M. Camerini, L. Fratta, and F. Maffioli, "The K best spanning arborescences of a network", Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Milano, Italy, preprint.

[E] S. Even, Algorithmic Combinatorics, Macmillan Co., New York, 1973.

[G] H. N. Gabow, "Two algorithms for generating weighted spanning trees in order", SIAM J. Comp. 6, 1977, pp. 139-150.

[HK] J. Hopcroft and R. Karp, "An $n^{5/2}$ algorithn for maximum matchings in bipartite graphs," SIAM J. Comp. 2, 1973, pp. 225-231.

[IRT] A. Itai, M. Rodeh, and S. L. Tanimoto, "Some matching problems for bipartite graphs", J.ACM, to appear.

[J75] D. B. Johnson, "Finding all the elementary circuits of a directed graph", SIAM J. Comp. 4, 1975, pp. 77-84.

[J77] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks", J.ACM 24, 1977, pp.1-13.

[JK] D. B. Johnson and S. D. Kashdan, "Lower bounds for selection in $X+Y$ and other multisets", SIAM J. Comp. 7, 1978, pp. 147-153.

[KIM] N. Katoh, T. Ibaraki, H. Mine, "An algorithm for finding K minimum spanning trees", Dept. of Appl. Math. and Physics, Kyoto Univ., Kyoto, Japan, preprint.

[L72] E. L. Lawler, "A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem", Mgt. Sci. 18, 1972, pp. 401-405.

[L76] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.

[M] K. G. Murty, "An algorithm for ranking all the assignments in increasing order of cost", Op.Res.16, 1968, pp. 682-687.

[MY] M. Machtey and P. Young, An Introduction to the General Theory of Algorithms, North-Holland Co., New York, 1978.

[RT] R. C. Read and R. E. Tarjan, "Bounds on backtrack algorithms for listing cycles, paths, and spanning trees", Networks 5, 1975, pp. 237-252.

[SL] J. L. Szwarcfiter and P. E. Lauer, "A search strategy for the elementary cycles of a directed graph", BIT 16, 1976, pp. 192-204.

[V77a] L. G. Valiant,"The complexity of computing the permanent", Int. Rep.CSR-14-77, Dept. Comp. Sci., Univ. of Edinburgh, Edinburgh,Scotland, Oct.1977.

[V77b] L. G. Valiant, "The complexity of enumeration and reliability problems", Int. Rep. CSR-15-77, Dept. Comp. Sci., Univ. of Edinburgh, Edinburgh, Scotland, Oct. 1977.

[Y] J. Y. Yen, "Finding the K shortest loopless paths in a network", Mgt. Sci. 17, 1971, pp. 712-716.

OPTIMAL ALGORITHMS FOR SENSITIVITY ANALYSIS IN ASSOCIATIVE MULTIPLICATION PROBLEMS

ARNIE ROSENTHAL
The University of Michigan, Computer and Communication Sciences Dept.
Ann Arbor, Michigan 48109

ABSTRACT

We consider efficient ways of determining the sensitivity of a product to changes in individual factors. The task is motivated by several interesting combinatorial and numeric problems which can be given a unified formulation -- namely, the problem of finding the (associative) product of N objects. Both deterministic and probabilistic changes to the factors are considered. Algorithms for two kinds of deterministic variation schemes are considered. Nontrivial lower bounds are obtained which demonstrate the algorithms to be optimal. For probabilistic choice of the parameter to be varied, it is shown that optimal ordered binary search trees or Huffman trees determine the optimal strategies. A number of unsolved problems are posed.

Keywords: Parameter variation, sensitivity analysis, unified algorithm, optimal algorithm, error correction, dynamic programming, resource allocation, equipment replacement, group knapsack, convolution, optimal, Lagrange interpolation, semigroup, associative multiplication.

CR Categories: 5.2, 5.3, 3.9, 8.3

Acknowledgements:

R.M. Karp, R. Rivest, W. Rounds, P. Helman, and M. Hersey all made helpful comments. This work was partially supported by a grant from Horace H. Rackham Graduate School and by National Science Foundation MCS77-01753.

1. Introduction

We consider a number of apparently different applications for which parameter variation studies are performed. By "parameter variation", we mean that each individual input will be changed, and the effect of this change on the problem solution noted.

Parameter variation studies (or equivalently, sensitivity analyses) are necessary in many optimization and other computational problems. When the input data is known to be unreliable, it is desirable to study how the optimal value changes if some of the data is altered. Such sensitivity analysis procedures may exploit duality [SAL] or linearity [GOT] in the problem domain, or properties of certain algorithms for graphical computations [CHE, CHI, SPI], or of certain branch and bound algorithms [GEO]. Nonserial associativity is exploited in [ROS1, ROS2]. [WAG] exploits the simple structure of regular languages.

We consider one-variable-at-a-time sensitivity analysis on a very simple kind of computation: multiplying N quantities $Z_1,\dots,Z_N$ in some arbitrary associative multiplication structure (i.e., semigroup). Given new values $Z'_1,\dots,Z'_N$, define variants by: $\text{Variant}_k \equiv (\prod_{i<k} Z_i)\, Z'_k (\prod_{i>k} Z_i)$. Variant_0 will denote $\prod_{i=1}^{N} Z_i$. The sensitivity analysis problem is:

Compute $\{\text{Variant}_k \mid k = 0,1,\ldots,N\}$.

Note that there is no scalar measure of sensitivity; rather, we simply output all the variants. The product operation is assumed only to be associative.

2. Applications

1) Error correction: If $a_1,\ldots,a_N$ is an input string of a finite automaton with transition function δ, δ_{a_1} can be defined as the transition function corresponding to character a_1. [That is, if q is a state of the machine, then $\delta_{a_i}(q) = (q,a_i)$].

We can then extend our definition so that $\delta_{a_j \ldots a_1}$ is the transition function corresponding to the string $a_j \ldots a_k$. Our goal is to compute the transition function corresponding to the input string $a_1 a_2 \ldots a_N$. The semigroup element Z_i is the one-step transition function δ_{a_i}.

The parameter variation problem is to study how the function $\delta_{a_1 \ldots a_N}$ is changed if a_k is changed to a'_k. The scheme could be useful in error correction attempts for regular languages. For instance, an error correcting interpreter could quickly find the "most likely" single-character errors. Attempts to insert semicolons as terminators, to interpolate multiple characters, or to transpose adjacent pairs of characters require only a small modification of our procedures. See [WAG] for a more powerful but more complex approach to this problem.

2) Consider optimization problems expressable as serial discrete decision processes, solvable by either forward or backard dynamic programming. A somewhat simplified version may be expressed as follows: Given N known functions, $f_1,\ldots,f_N$, each a function of two variables with finite ranges, compute the function

$$h(x_1,x_{N+1}) \equiv \min_{\substack{\{\text{values of}\\ x_2,x_3,\ldots,x_N\}}} [f(x_1,x_2) + f_2(x_2,x_3) + \ldots + f_N(x_N,x_{N+1})].$$

The parameter variation problem is to explore the effect of varying each f_i. Each Z_i represents a function f_i. $Z_i Z_{i+1}$ is the function g given by

$$g(x_i,x_{i+2}) = \min_{\{x_{i+1}\}} [f_i(x_i,x_{i+1}) + f_{i+1}(x_{i+1},x_{i+2})].$$

The next three problems may be transformed into instances of the above optimization problem. The state variables of the usual dynamic programming recurrences map to the variables x_i in application 1 above. The examples are included because they are more familiar than the general optimization problem. Ref [ROS-3] discusses some implementation considerations.

3) Assignment of tasks to machines: Suppose a task can be decomposed into N sequential stages. Doing stage i at location j costs $p_i(j)$. Moving the output of stage i from location j to k costs $C_i(j,k)$. We wish to choose $\{x_i \overset{\Delta}{=} \text{location of task } i \mid i = 1,\ldots,N\}$ to minimize the total processing costs plus communications costs. That is, minimize

$$\sum_{i=1}^{N-1} [p_i(x_i) + c_i(x_i,x_{i+1})] + p_N(x_N).$$

4) Consider the classical resource allocation problem of allocating limited resources among competing projects so as to maximize profit. The problem may be formalized as: Given functions $r_j(.)$, $j = 1,\ldots,N$, for each value of TOTAL, $0 \leq \mathrm{TOTAL} \leq \mathrm{LIMIT}$ (for some known limit), choose the single stage allocations $Y_1, Y_2, \ldots, Y_N$ to maximize $\sum_1^N r_j(Y_j)$ subject to $\sum Y_j = \mathrm{TOTAL}$.

The functions r_j are the inputs that are to be varied, and the maximum value of $\sum_1^N r_j(Y_j)$ is the output. $f_j(x_j, x_{j+1}) \equiv r_j(x_{j+1} - x_j) \cdot x_1$ is fixed at 0. $h(0, x_{N+1})$ tells the best return achievable with x_{N+1} resource units. The parameter variation problem is to explore the effect of altering each f_j.

5) Equipment replacement: A machine's operating cost changes with time. At the start of each time period, one may exchange a machine of any age for a machine of any other age. (A cash payment may be involved in the trade.) Functions f_i which tell the trading plus operating cost during period i for each possible trading action are knwon. The problem is to choose a replacement strategy to minimize costs over some given number of time periods. The parameter variation problem is to explore the effect of varying each f_i.

6) The group knapsack problem, which is fully stated in [SAL], is a kind of multidimensional resource allocation problem which arises in relaxation methods for integer programs. One seeks to maximize CX while preserving the condition AX = B in the appropriate ring. The parameter variation problem is to study the effect of changing columns of A. The semigroup element Z_i is the i^{th} column vector in the ring.

7) Multiplication of N matrices: Given equal sized square matrices $M_1, \ldots, M_N$, find the product $M_1 M_2 \ldots M_N$. The parameter variation problem is to explore the effect of changing each M_i.

8) Convolution of N functions: The probability density function of the sum of N random variables is formed by the convolution of the individual densities. In one application (noted by D. Shier) the individual random variables represent sources of measurement error, and the probability density function for total error is desired. The parameter variation problem is to find the effect of altering the individual densities. The semigroup element Z_i is the density function of the i^{th} random variable.

9) In many problem domains, one needs to find the N products of N-1 out of N factors, once for each choice of the factor to be omitted. (The terms of a Lagrange interpolating polynomial can be found by computations of this form.)

The factors are the inputs to be varied (i.e., replaced one at a time by an identity) and the product of the factors is the output. The semigroup elements may be polynomials of the form $(x - a_i)/(a_j - a_i)$.

10) Multiplication of N polynomials: (either ordinary polynomials, or polynomials in a quotient field)

3. A simple, yet optimal algorithm

Actually, devising sensitivity analysis algorithms is difficult only in the specific settings. (The trade-in policy problem was our starting place.) Algorithm design is much easier when one abstracts to a semigroup setting.

The most direct solution to the sensitivity analysis problem is to compute each variant separately. N-1 multiplications will be needed for each k, for a total of $O(N^2)$. With a little care, the running time can be reduced to O(N).

Define left partial products L_i for $1 \leq i \leq N$, $L_i \triangleq \prod_{j \leq i} Z_j$. Also, define right partial products $R_i \triangleq \prod_{j \geq i} Z_j$. We adopt the convention that multiplying any X by the empty terms L_o or R_{N+1} results in $\underline{X}$ but costs nothing.

Algorithm 1.

```
1) Multiply the terms $Z_i$ right to left, obtaining
       $\{R_i \mid i=N-1,\dots,1\}$
2) Multiply the terms $Z_i$ left to right, obtaining
       $\{L_i \mid i=2,\dots,N-1\}$
3) For k = 1 until N:  compute $\text{Variant}_k := L_{k-1} Z'_k R_{k+1}$
```

Complexity. Exactly (N-1) + (N-2) + (2N-2) = 4N-5 multiplications are performed. (N-2) intermediate results must be stored.

It seems desirable to relate the multiplication count to running time in each application. For "number of algebraic multiplications" to be a useful complexity measure, the time for multiplications must be roughly constant. For certain applications which were deliberately omitted from our list (e.g., the 0-1 knapsack problem), an elementary term Z_k (which in knapsack problems has only two entries) requires much less time to multiply than a partial product. In this kind of situation, formula 1 (whose N^2 multiplications all have a simple term Z_i for one operand) may be more attractive than our algorithm.

4. Optimality and near-uniqueness

We will obtain a lower bound on the algorithm's worst case performance by considering its behavior in a situation without useful input-dependent relations, e.g., the free semigroup with unequal inputs. Algorithms operating in this setting are essentially straight line programs, performing a sequence of semigroup multiplications. To provide a handier description, we model an arbitrary straight line program on given size input by a circuit. Inputs to the circuit are the values $Z_1,\dots,Z_N$, $Z'_1,\dots,Z'_N$. Each gate forms the product of its two inputs. The circuit's outputs are the desired variants. A circuit is optimal if it contains the minimum possible number of multipliers.

Theorem 1 (below) shows that Algorithm 1 is optimal. Theorem 2 shows that all optimal algorithms are nearly the same as Algorithm 1.

Theorem 1. Every algorithm for the sensitivity analysis problem uses at least 4N-5 multiplications in the worse case(i.e., Algorithm 1 is optimal).

Proof. We show that there is an optimal circuit which includes all of the multiplications performed by Algorithm 1; this will immediately imply that the circuit is the circuit of Algorithm 1.

Consider any optimal circuit, and let P_k denote the path in the circuit from Z'_k to Variant_k.

Lemma 1. Any optimal circuit may be transformed into an optimal circuit in which

i. L_{k-1} and R_{k+1} are available as multiplier outputs

ii. The output values are unchanged for all multipliers not on P_k in the original circuit.

Proof of lemma. First, a transformation will be defined. We begin by deleting P_k from the circuit. Next, insert new multipliers in such a way that all the orphaned multipliers off P_k which were inputs to P_k from the left [right] are multiplied together. Now, multiply the grant left product by Z'_k and the result by the grant right product. It is shown below that the transformed circuit still solves the sensitivity analysis problem (i.e., computes all the variants), still uses a minimum number of multiplications, and satisfies conditions i and ii.

We begin with some observations. Since the original circuit was optimal, it could not compute any useless outputs. Any output which included the same term twice, or else two "primed" terms would be useless, and hence not present. Thus, the union of all paths below Variant_k must be a tree, with Z'_k the only primed input. Finally, all trees on N leaves have the same number of multipliers (i.e., internal nodes).

It is readily seen that multipliers on P_k are used only to compute Variant_k and that no multipliers of P_k have been altered (although the destination of some outputs may have changed). Thus, variants other than Variant_k are unaffected by the transformation. Now, the grant product formed on the left includes all terms to the left, and hence is L_{k-1}. Similarly, the grand right product is R_{k+1}. Multiplying these with Z'_k as specified produces Variant_k. Hence, the circuit still computes all the variants. This argument has also verified that conditions i and ii of the lemma still hold.

The computation of Variant_k may still be modelled by a tree with N leaves, and hence uses the same number of multiplications. Hence, the new circuit is still optimal, and the lemma is proved.

Proof of theorem continued. Because paths $P_{k'}$ and $P_{k''}$ are disjoint for $k' \neq k''$, and because the transformation only changes multiplier outputs along the path being transformed, it is possible to apply the transformation of Lemma 1 for different values of k without interference. If one starts with an arbitrary optimal circuit and applies the transformation for every k, k = 1,...,N, one obtains an optimal circuit which has available all the variants, plus $L_{k-1}Z'_k$, and R_{k+1} for all k (assuming suitable interpretations for the cases of k = 1 and N). But these are all the multiplier outputs of Algorithm 1; hence, that circuit is exactly the circuit of Algorithm 1, and Algorithm 1 is optimal.

The following theorem implies that any optimal circuit must be essentially the same as the circuit for Algorithm 1. The circuits will, in fact, differ only in the association of the last two multiplications along the path to each Variant_k $(k > 0)$, and in the value of i used in the multiplication which computes $\text{Variant}_0 = L_iR_{i+1}$.

Theorem 2. Every optimal circuit computes$\{ L_i \mid i=2,\dots,N-2\}$, $\{R_i \mid i=3,\dots,N-3\}$ and for each i $(i=2,\dots,N-1)$ one of $L_{i-1}Z'_i$ or Z'_iR_{i+1}. No optimal circuit computes $Z_iZ'_j$, Z'_iZ_j, or Z_iZ_j unless $j=i+1$ and either $i=1$ or $j=N$.

Proof. If $j \neq i+1$, the product Z_iZ_j would not appear in any otuput, and could be deleted. A multiplication will be called internal if it is of the form Z_iZ_{i+1} where $i\neq 1$ and $i+1\neq N$. Note that the sequence of transformations from Lemma 1 do not alter any internal multiplications, but that the resulting circuit (Algorithm 1) includes no internal multiplications. The conclusion is that the arbitrary optimal circuit at the start of the transformations also contained no internal multiplications.

We are now prepared to show that for each i ($i=2,\ldots,N-2$) L_i must be computed by every optimal circuit. Apply the transformation of Lemma 1 to the path P_{i+2} which computes Variant_{i+2}. The result is a new optimal circuit containing L_{i+1}. Internal multiplications are forbidden to optimal circuits, so L_{i+1} must be computed by L_iZ_{i+1}, i.e., L_i is present in the transformed circuit. If the claim of the theorem is false for i, L_i must be the new output of a multiplier newly created (by the transformation), and hence not an input to P_{i+1}. Now, apply the transformation to P_{i+1}; the newest circuit will have a multiplier whose output is L_i inputting into Z'_{i+1}. A paradox has been achieved -- the circuit is still optimal, but it has two multipliers with the same output (L_i). The only resolution to the paradox is that L_i must have been the value at some multiplier in the original circuit. It is easy to show that as long as L_i is available, it will be suboptimal to compute $Z_iZ'_{i+1}$ since at the same cost one can compute $L_iZ'_{i+1}$.

The argument about $\{R_i\}$ and $\{Z'_iZ_{i+1}\}$ is analogous.

Q.E.D.

5. Other kinds of sensitivity analysis

In this section, a number of results for other sensitivity analysis problems will be presented. All the problems (including the one above) can be considered as the execution of a series of commands of the form CHANGE Z_i to Z'_i, RESTORE Z_i to its previous value, and compute PRODUCT. (The command handler may wish to store some intermediate products.) In the sensitivity analysis problem presented above, each CHANGE command was immediately followed by PRODUCT and RESTORE, each Z_k was CHANGEd exactly once, and the entire sequence was known in advance. There are several other interesting kinds of problems having different restrictions and different advance information.

5.1. In a hill-climbing optimization procedure, one might not RESTORE Z'_i to its original value before altering Z_{i+1}. If RESTOREs are omitted, another interesting sensitivity analysis problem is obtained. An interesting (open) problem could be to optimally handle arbitrary sequences of CHANGEs. We consider in this section only the case, where CHANGEs are made successively to $Z_1, Z_2, \ldots$ (In commutative semigroups, if the sequence of CHANGEs is known in advance and no term is CHANGEd twice, one may reorder terms to achieve this situation.)

As before, let Variant_0 denote $\prod_{i=1}^{N} Z_i$. Let $\text{Variant}_k \triangleq (\prod_{j<k} Z'_j)(\prod_{j>k} Z_j)$. The sensitivity analysis problem of this section is again to compute $\{\text{Variant}_k \mid k=0,\ldots,N\}$.

A simple algorithm is optimal for this problem too. Let $R_k \triangleq \prod_{i \geq k} Z_i$ (as before), and let $L'_k \triangleq \prod_{i \leq k} Z'_i$.

1. Compute $\{R_k \mid k=N-1,N-2,\ldots,1\}$ by multiplying right to left.
2. Compute $\{L'_k \mid k=2,\ldots,N\}$ by multiplying left to right.
3. For k=1 until N compute $\text{Variant}_k := L'_k R_{k+1}$ (by convention, multiplying by the nonexistent R_{N+1} has no effect or cost)

This algorithm requires 3N-3 multiplications, and storage for N-2 intermediate results.

Theorem 3. Algorithm 2 provides an optimal circuit for the second sensitivity analysis problem. This optimum circuit is unique.

The proof will apear in a later version of this paper. The proof seems to imply a stronger result: Algorithm 2 has the unique optimal circuit for solving the second sensitivity analysis problem in commutative semigroups. This stronger form is interesting, because several applications involved a commutative multiplication operation.

5.2. In previous examples, we knew in advance what variables were to be CHANGEd, i.e., what variants were to be computed. Suppose now that after the initial product is computed, a user at a terminal gives commands "CHANGE Z_i and compute PRODUCT" without any prespecified ordering. Products must be computed immediately, i.e., on line. We will assume that N-2 intermediate results may be stored (all the intermediates from one computation). How may the number of multiplications be kept small? An elegant solution is obtained by reformulating the problem using binary trees.

Left-to-right multiplication is just one way of associating (i.e., parenthesizing) the elements Z_i to form the product. Any association may be represented by a binary tree, with the values Z_i at the leaves and each internal node representing the product of its two sons. The original product $\prod Z_i$ is computed according to some binary tree, and the intermediate results are stored. Now, suppose Z_k is CHANGEd to Z'_k. It will be necessary to recompute values in the tree along the path from Z_k to the root; the number of new multiplications equals the length of that path.

5.2.1. Optimal worst case

To minimize the worst-case time, the multiplication pattern should use a uniform binary tree. With such a tree, at worst $\lceil \log_2 N \rceil$ CHANGEs will be needed. By contrast, the tree corresponding to left-to-right multiplication has worst-case path length N-1.

$\lceil \log_2 N \rceil$ is optimal, as long as only one set of intermediate results (i.e., only one tree) is allowed to be stored. We conjecture that keeping more than one tree would be counterproductive in practice, as the time to update all trees after unrestored CHANGEs would probably exceed the time saved.

5.2.2. Optimal expected number of multiplications

Let the weight of the ith leaf node be the frequency with which Z_i is CHANGEd. To minimize expected number of multiplications, we need the ordered binary tree with minimum weighted external path length. T.C. Hu has a complicated O(N log N) algorithm for this problem [KNU].

In many applications, multiplication commutes, so it is undesirable to enforce an ordering on the leaves of the tree. To minimize the expected number of multiplications with reordering permitted, one uses a Huffman tree (which is found by a very simple O(N log N) algorithm [KNU].

5.3.1. Off-line computation

Suppose the sequence of all future commands is known in advance, but is not of the simple forms considered in sections 3 or 5.1. This problem is open. A greedy strategy may be appropriate.

5.3.2. Unknown stationary random process

No knowledge of the relative frequencies is available, but the process generating CHANGEs is known to be stationary, so that relative frequencies can be estimated. This is also open. The results in [ALL] about adaptive trees may be relevant.

5.3.3. Multiple CHANGEs

All problems considered so far have command lists with at most one CHANGE between PRODUCT commands. This restriction might be relaxed.

5.3.4. Commutativity

If multiplication commutes, then the lower bound argument of Theorem 1 is no longer valid, since it assumes Z'_k must be multiplied separately on the left and right. We know neither a way to resurrect the lower bound nor a way to speed the algorithms by exploiting commutativity.

(If multiplication commutes and has an inverse (i.e., the semigroup is an Abelian group), then our algorithm is uninteresting because it is sufficient to multiply P by $Z'_k Z^{-1}_k$ to CHANGE Z_k to Z'_k. To solve our original problem, (N-1) + 2N = 3N-1 multiplications plus N inverse computations are needed.)

5.5. Time-space tradeoffs

The naive $O(N^2)$ algorithm requires storage for only one intermediate product; a slightly modified version of the naive scheme uses 1/2(N-1)(N+2) multiplications and two intermediate results. Our time-optimal method used 4N-5 multiplications and N-2 locations. We suspect that for this problem, the time-space product must be $O(N^2)$. An interesting research question would be to find the optimal algorithm as a function of alloted storage.

6. Summary

The study of associative multiplication problems has been motivated. For several kinds of sensitivity analysis in associative multiplication problems we have presented algorithms and demonstrated the algorithms' optimality. A number of open questions have been posed.

REFERENCES

[ALL] B. Allen and I. Munro, "Self Organizing Binary Search Trees", F.O.C.S. Proceedings, pp. 166-172, 1976.

[CHE] G.A. Cheston, Incremental Algorithms in Graph Theory, Technical Report 91, Department of Computer Science, University of Toronto, 1976.

[CHI] F.Y. Chin and D. Houck, "Algorithms for Updating Minimum Spanning Trees", J.C.S.S. 16 3(June 1978), pp. 333-344.

[GEO] A.M. Geoffrion and R. Nauss, "Parametric and Postoptimality Analysis in Integer Linear Programming", Management Science 23 5(Jan. 1977), pp. 453-466.

[GOT] S. Goto and A. Sangiovanni-Vincentelli, "A New Shortest Path Updating Algorithm" (to appear in Networks).

[KNU] D. Knuth, The Art of Computer Programming, Addison-Wesley, Reading, Mass., 1973.

[MUR] K. Murty, Linear and Combinatorial Programming, John Wiley, N.Y., 1976.

[ROS-1] A. Rosenthal, "Sensitivity Analysis in Nonserial Optimization Problems", to be submitted for publication.

[ROS-2] A. Rosenthal, "Decomposition Algorithms and Sensitivity Analysis for Probabilistic Circuits and Fault Trees", submitted for publication.

[ROS-3] A. Rosenthal, "Sensitivity Analysis in Serial Dynamic Programming Problems", submitted to Management Science.

[SAL] H. Salkin, Integer Programming, Addison-Wesley, 1975.

[SPI] P. Spira and A. Pan, "On Finding and Updating Spanning Trees and Shortest Paths", SIAM J. Comp. 4(3), 1975, pp. 375-380.

[WAG] R. Wagner, "Order-n Correction for Regular Languages", CACM 17(5), May 1974, pp. 265-268.

AN OPTIMAL SCHEME FOR ADDRESSING IN DIGITAL LOOP SYSTEMS

JATINDER P. BHULLAR
Dept. of Elec. Engineering &
Communications Research Lab.
McMaster University
Hamilton, Ont.
Canada L8S 4L7

JOHN B. ANDERSON
Visiting Associate Professor
Dept. of Elec. Engineering &
Computer Science
Univ. of California at Berkeley
Berkeley, CA 94720 U.S.A.

ABSTRACT

Addressing is a basic requirement in most network types. The transmitted addresses are used in a variety of ways. User identification, dynamic assignment of capacity, and network control are few examples. A new scheme for addressing in a digital loop is derived. An encoding algorithm and its inverse are presented. The new scheme is evaluated in comparison with other addressing techniques.

1. INTRODUCTION

A digital loop consists of a set of n users joined by a serial digital communication channel. Each user in the loop can observe and react with the information flow in the channel. The user interaction constitues the transfer of information in the loop. Digital loop networks provide an approach to handling intermittent and quick response traffic. Fig. (1) illustrates a typical loop configuration. The data formats represent two different loop implementations. In a Pierce loop [1], data is read into empty blocks and read out from full blocks. In an alternate arrangement of Newhall [2], a control word is generated by a supervisor station. Each terminal appends its information to the message frame as it receives the control word. The loop cycle recommences as the control word reaches back to the supervisor station. A scheme proposed by Graham and Pollak [3] is easier to implement though not as efficient.

We propose a new optimum scheme for addressing in a digital loop. It requires moderately complex encoding and decoding but uses the least possible number of bits for transmitting address information. Our investigation was prompted in part by papers on source coding by Schalkwijk [4] and Cover [5]. In [4] and [5] schemes for indexing binary sequences under different constraints are investigated. The underlying principle is to send the "index" of an n-bit sequence. The transmitted index requires less than n bits in general and thus data compression is achieved. Our scheme is based on a new algorithm which generates a "composite index" for a given digital sequence. We identify the relationship between efficient addressing in an n-terminal loop and the generation of composite index for an n-bit sequence.

Thus results to be obtained in section (2) become directly applicable to digital loop addressing. Since it will be shown that composite indexing provides optimal data compression, the addressing scheme so generated is also optimal. Hence we could use the technique in a digital loop environment for efficient polling, data formatting, etc.

2. COMPOSITE INDEX OF A SEQUENCE AND ITS WEIGHT

We derive the essential mathematical relationships in this section. Basic theorems are proved leading to the optimal composite indexing scheme. We describe a procedure to evaluate composite index and its inverse for an n-bit sequence.

2.1 Basic Theorems

Consider a binary sequence $u = (u_1, \ldots, u_n)$ where $u_i \ \epsilon \ \{0,1\}$. We take the usual definition for the weight of a sequence to mean the total number of non-zero components in the sequence. Let $r(u)$ denote the rank of a sequence u such that $r(010{-}{-}{-}0) > r(100{-}{-}{-}0)$. We note that the ranking is inverse of the ordinary lexicographic sense. It has been designed as such to facilitate implementation of the scheme in a digital loop environment. We denote by $U(n,w)$ the set

$$\{\{0,1\}^n ; \ \sum_{i=1}^{n} u_i = w\} .$$

Definition 1: The rank of a binary sequence $u \ \epsilon \ U(n,w)$ is given by

$$r(u) = \sum_{k=1}^{n} u_k \binom{k}{w(k)-1} \tag{1}$$

where

$$w(k) = \sum_{j=1}^{k} u_j \leq n \quad \text{and} \quad \binom{n}{w} = 0 \quad \text{for } w < n.$$

Then we have the following theorem.

Theorem 1: The rank $r(u)$ of a sequence as defined in (1) is unique.

Proof: An outline of the proof will be given; for a complete proof see [4].

The proof follows by induction on n. With $n = 1$, we can have $w = 0$ or $w = 1$. From (1), $r(0) = 0$, since $u = 0$ is the only allowed sequence for w

= 0. Similarly for $w = 1$, $u = 1$ is the only allowed sequence and from (1), $r(1) = 0$. Suppose that the theorem holds for each $u(n,w)$, $0 \leq w \leq n$, with $n \leq N$. We will show that the theorem holds for $U(N+1,w)$, $0 \leq w \leq N+1$. Consider the set $U_o(N+1,w) = \{u \in U(N+1,w);\ u_{N+1} = 0\}$. By the inductive hypothesis

$$r_o(u) = \sum_{k=1}^{N} i \binom{k}{w(k)}$$

Similarly for set $U_1(N+1,w)$ we have

$$r_o(u) = \sum_{k=1}^{N} u_k \binom{k}{w(k)}$$

The number of elements of $U_o(N+1,w)$ is $\binom{N}{w}$ and $0 \leq u_o(t) \leq (\binom{N}{w}-1)$. Thus the total set $U(N+1,w)$ can be ranked according to

$$r(u) = (1-u_{N+1})\ r_o(n) + u_{N+1}\ [\binom{N}{w} + r_1(u)] = \sum_{k=1}^{N+1} \binom{k-1}{w(k)}$$

This completes the proof.

We define a more useful quantity called composite index.

Definition 2: The composite index $i(u)$ for a sequence $u = (u_1, \ldots, u_n)$ is given by,

$$i(u) = \sum_{k=1}^{n} u_k \left\{\binom{k-1}{w(k)} + \binom{n}{w(k)-1}\right\} \qquad (2)$$

Theorem 2: The composite index as defined in equation (2) is unique.

Proof: We need the following lemma to complete the proof.

Lemma 1: The term

$$\sum_{k=1}^{n} u_k \binom{n}{w(k)-1}$$

in (2) is a constant for all sequences in $U(n,w_1)$.

Proof of lemma 1: We note that for sequences belonging to $U(n,w_1)$

$$\sum_{k=1}^{n} u_k \binom{n}{w(k)-1} = 1 + \binom{n}{1} + \ldots + \binom{n}{w_1-1} = \sum_{i=0}^{w_1-1} \binom{n}{i} \qquad (3)$$

The term given by (3), insures the unique partitioning of the space of all n-sequences into sequence sets of equal weight.

Applying theorem (1) and lemma (1) to (2), concludes the proof for theorem (2).

Theorem 3: If w is less than 0.33 n, then the composite index encodes the weight and rank of a sequence u(n,w) such that

$$\log_2 \binom{n}{w} < \log_2 i(u) < \log_2 \binom{n}{w} + 1$$

where $\log_2$ i(u) is the average number of bits needed to encode the composite index.

Proof: The proof of the theorem depends upon the following lemma.

Lemma 2: The following are true for w = n/3,

$$\binom{n}{w} > \sum_{i=0}^{w-1} \binom{n}{i} \tag{4a}$$

$$\binom{n}{w+1} < \sum_{i=0}^{w} \binom{n}{i} \tag{4b}$$

Proof of lemma 2:

Let

$$\sum_{i=0}^{w-1} \binom{n}{i} = \sigma$$

For (4a) we have

$$\binom{n}{w} > \sigma \tag{5}$$

Now,

$$\binom{n}{w+1} = \frac{n!}{(w+1)!(n-w-1)!} = \binom{n}{w} \frac{n-w}{w+1}$$

But,

$$\sum_{i=0}^{w} \binom{n}{i} = \binom{n}{w} + \sigma$$

Thus from (4b) we get

$$\binom{n}{w} \frac{n-w}{w+1} < \binom{n}{w} + \sigma \tag{6}$$

Using the relation (4a) in (6) and simplifying we have

$$w = \left[\frac{n}{3}\right] + 1 \tag{7}$$

where [x] means integer part of x.

Considering large n and w completes the proof of the lemma.

Now from (2)

$$i(u) = \sum_{k=1}^{n} u_k \binom{k-1}{w(k)} + \sum_{k=1}^{n} u_k \binom{n}{w(k)-1}$$

It is easy to show that for a sequence with Hamming weight w,

$$\sum_{k=1}^{n} u_k \binom{n}{w(k)-1} = \sum_{i=0}^{w-1} \binom{n}{i} \tag{8}$$

Similarly from theorem [1], we note that the maximum rank for such a sequence is given to be

$$r_{max}(u) = \binom{n}{w} - 1 \tag{9}$$

Using (8) and (9) in (7) we see that

$$i_{max}(u) = \binom{n}{w} - 1 + \sum_{i=0}^{w-1} \binom{n}{i}$$

Applying (4a) and lemma (2) gives

$$i(u) < 2 \binom{n}{w} \tag{10a}$$

From theorem [2] we know that

$$i(u) > \binom{n}{w} \tag{10b}$$

Combining (10a), (10b) and taking logs proves theorem (3).

In the analysis given above we have considered only binary sequences. The results can be generalized to q-ary sequences with some manipulation. The proofs follow immediately from that of the binary case.

2.2 Composite Index Computation

We know from thoerem (2) the composite index of a sequence $u = (u_1, \ldots, u_n)$ is given by,

$$i(u) = \sum_{k=1}^{n} u_k \left\{ \binom{k-1}{w(k)} + \binom{n}{w(k)-1} \right\}$$

Thus the computation of the index involves finding the weighted sum of binomial co-efficients. In the binary case, when the weight w of the sequence is much less than n, the index calculation requires determining a few binomial co-efficients. Using the following basic recurrence relationship for binomial co-efficients, the calculation is straight forward.

$$\binom{m}{r} = \binom{m-1}{r-1} + \binom{m-1}{r} \tag{11}$$

The equation (11) leads to the famous Pascal's triangle construction for sequential generation of bionomial co-efficients. A similar construction has been exploited by Lynch [6] for data compression.

Now we consider computing the sequence from a given composite index or the inverse transformation. The algorithm is given below.

```
STEP 1
        x ← y ← i(u);
        j ← 0
        while y > 0 do
              begin
                    y ← x - $\binom{n}{j}$;
                    j ← j+1;
              end
        ;
        w ← j-1;
        i(u) ← y + $\binom{n}{w}$;
STEP 2
        z ← z1 ← i(u);
        m ← 1;
        for k=1 to w do
              begin
                    for $\ell$ = m to n do
                          begin
                                z1 ← z - $\binom{n-\ell}{w-k+1}$;
                                if z1 > 0
                                then
                                      begin
                                            z ← z1;
                                            $u_{(n-\ell+1)}$ ← 1;
                                      end
                                else $u_{(n-\ell+1)}$ ← 0;
                          end
                    ;
              end
        ;
```

Thus we see that step 1 determines which subset of n-bit sequences does i(u) belong to. At the end of step 1, evaluation is reduced to determining which one out of $\binom{n}{w}$ combinations is present.

3. COMPOSITE INDEXING RELATED TO LOOP ADDRESSING

In this section we apply the composite indexing scheme to the loop addressing environment. We also briefly describe some of the other addressing techniques used for polling and data transmission.

3.1 Composite Index Loop Addressing

We consider the digital loop shown in Fig. (1). The terminals are arranged around the loop and are sequentially numbered using the consecutive integers from 1 to n. The numbers thus correspond to the terminal addresses. The central controller/supervisor may perform a variety of functions. Some typical tasks being handling network protocals, dynamic assignment of capacity, provision of timing signals, loop monitoring, etc. Address signalling in such a loop can be conveniently modeled as in Fig. (2). The flags set in positions 2 and (n-1) indicate that terminals 2 and (n-1) are transmitting their addresses for recognition. It becomes immediately clear from the same figure that address recognition is equivalent to the identification of one's in an n-bit sequence. Thus we establish the equivalence between binary sequences and loop addressing.

Now consider a typical case for binary data compression. Say the source generates one's with probability p and zero's with probability q=1-p. If $p << q$, the source output becomes a highly redundant data stream with source entropy << 1. In such a situation, data compression can significantly reduce the bit rate. The situation in digital loop addressing is strongly analogous. We know that the terminals make intermittent demands. In addition, only a small fraction out of n require service at any given instance. Thus reviewing Fig. (2), one can see how composite indexing scheme can be used for efficient addressing.

The scheme is carried out using the procedures set up in section 2.2. For encoding, the k-th terminal determines the number of users w already present in the address block. This is achieved by following Step 1 of the procedure. The binomial co-efficients $\binom{k-1}{w+1}$ and $\binom{n}{w}$ are computed and added to the address block. Complete decoding at the central controller is performed by following Steps 1 and 2 of the procedure. We see that for such an implementation, encoding and decoding of addresses will be moderately complex.

3.2 Comparison with Other Addressing Schemes

In discussing the schemes that follow, we will carry over the notation from the previous sections. Namely, we consider a set of n terminals out of which w signal at any given time.

One of the easiest ways of transferring addresses follows immediately from Fig. (2). Let a frame n bits long circulate the loop. A terminal communicates its address by setting up the appropriate flag in the frame. This is called the marker setting scheme. The scheme is very inefficient when w is much less than n. The scheme is optimal when an average of (n/2) addresses are signalled per frame.

Address concatenation scheme assigns an address of $\log_2 n$ bits to each terminal. The w addresses are communicated using $w\log_2 n$ bits on the average. Thus the average number of bits in the address frame increases linearly with w. It becomes less efficient than marker setting for w greater than $(n/\log_2 n)$.

An addressing scheme can also be devised using Davisson's [7] procedure for source coding. The scheme makes use of the result similar to theorem (1). It would use $\log_2 w\binom{n}{w}$ bits on the average for transmitting w out of n addresses. We can compare it with less than $\log_2 2\binom{n}{w}$ bits required by composite indexing. Thus it may be considered as a less efficient version of our scheme.

We compare the different addressing schemes in Fig. (3). It is clear that marker setting is least efficient for the practical values of n and w. The composite indexing scheme is optimal in the same region. The concatenation scheme and the scheme generated from [7] are bounded by the optimal and the least efficient schemes.

4. CONCLUSIONS

We have devised a scheme for addressing in a digital loop, using source coding ideas. This was made possible by virtue of the equivalent information conveyed in the two processes. We have compared our scheme with other techniques.

It is suggested that future work could involve providing different alphabet sizes to each of the n users. It could lead to an integrated information network since provision of limited intelligence at the terminal level is becoming cheaper. Thus if economical trade-offs can be justified, more efficient networks would result.

REFERENCES

[1] J.R. Pierce, "Network for Block Switching of Data", BSTJ, vol. 51, No. 6, July-August 1972, pp. 1133-1145.

[2] W. Farmer and E. Newhall, "An Experimental Distributed Switching System to Handle Bursty Computer Traffic", ACM Symposium on Problems in the Optimization of Data Communications, Pine Mountain, Georgia, October 1969.

[3] R.L. Graham and H.O. Pollak, "On the Addressing Problem for Loop Switching", BSTJ, vol. 50, No. 8, October 1971, pp. 2495-2519.

[4] J. Pieter M. Schalkwijk, "An Algorithm for Source Encoding", IEEE Trans. on Information Theory, vol. IT-18, No. 3, May 1972, pp. 395-399.

[5] T.M. Cover, "Enumerative Source Coding", IEEE Trans. on Information Theory, vol. IT-19, No. 1, January 1973, pp. 73-77.

[6] T.J. Lynch, "Sequence Time Coding for Data Compression", Proc. IEEE (Letts.), vol. 54, pp. 1490-1491, October 1966.

[7] L.D. Davisson, "Comments on 'Sequence Time Coding for Data Compression'", Proc. IEEE (Letts.), vol. 54, p. 2010, December 1966.

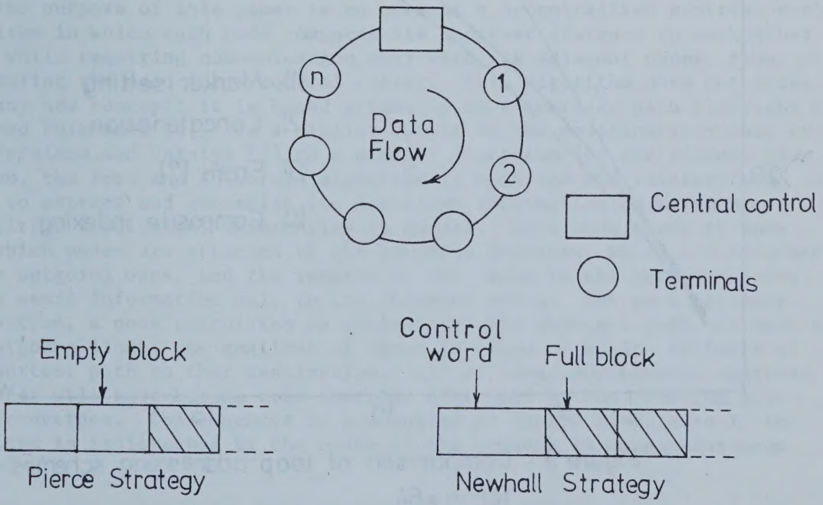

n
1
2
Data
Flow
Central control
Terminals
Empty block
Control word
Full block
Pierce Strategy
Newhall Strategy

Figure 1. Communication via a Digital Loop

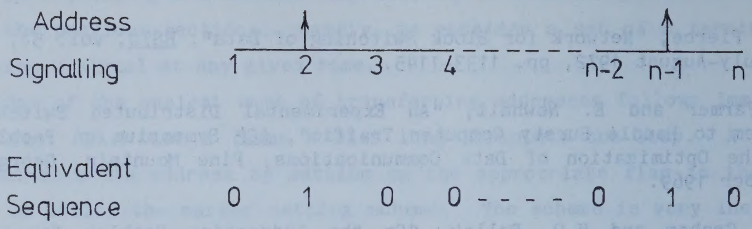

Address
Signalling
1
2
3
4
n-2
n-1
n
Equivalent
Sequence
0
1
0
0 - - - - 0
1
0

Figure 2. Binary information in addressing

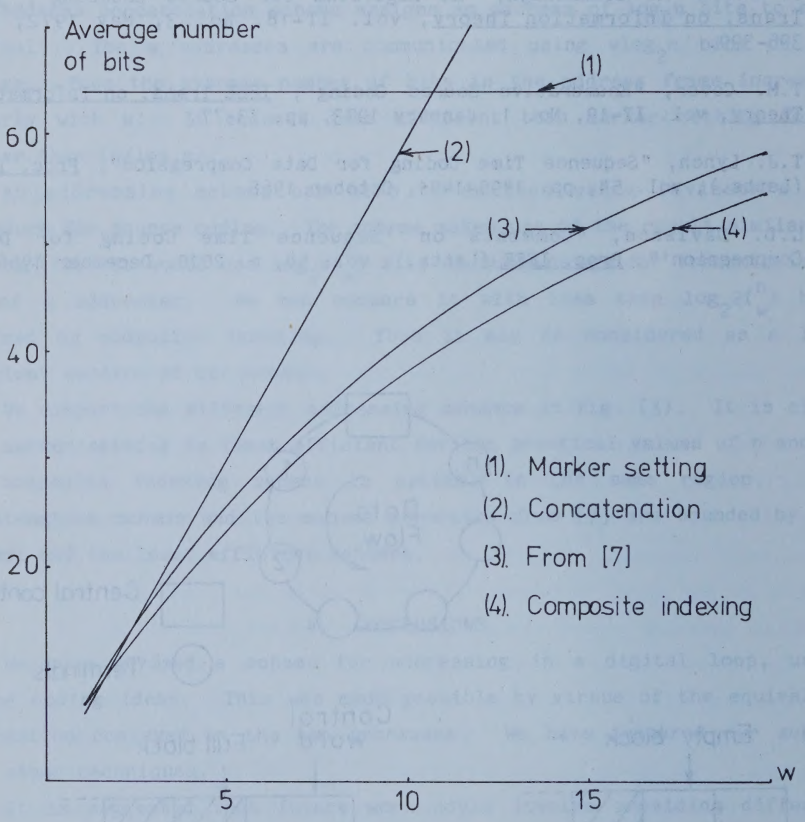


Figure 3. Comparison of loop addressing schemes for n = 64

A DECENTRALIZED SHORTEST PATH ALGORITHM

JEFFREY M. ABRAM and IAN B. RHODES
Department of Systems Science and Mathematics
Washington University, St. Louis, Missouri 63130

ABSTRACT

A decentralized algorithm for determining the shortest paths in a network is presented. Using information received only from neighboring nodes, a sequence of additions and comparisons is performed at each node in the network. Convergence to the optimal solution takes place in finite time.

I. INTRODUCTION

A common problem in graph theory is that of finding the shortest paths between all pairs of nodes in a network, and numerous algorithms exist for its solution, e.g. [1], [2], [3]. Nearly all of these algorithms are developed under the presumption that the computations will be performed by a decision maker with knowledge of the entire graph topology and of all branch lengths. The implementation of algorithms of this type can be thought of as requiring each node to transmit distance and topology information to a central controller, who is then responsible for solving the problem. After the shortest paths have been determined, the controller will send the appropriate routing information to each of the nodes. In a large network this could involve a significant amount of communication. Additionally, for some networks establishment of a central controller may be expensive, infeasible, or undesirable from a security or reliability viewpoint.

The purpose of this paper is to present a decentralized shortest path algorithm in which each node computes its shortest distance to each other node, while requiring communication only with its adjacent nodes, thus eliminating the need for a control center. This algorithm does not arise from any new concept; it is based primarily on a shortest path algorithm of Ford and Fulkerson [4]. In a similar spirit to the modifications made by Lau, Persiano and Varaiya [5] to a similar algorithm for the maximum flow problem, the Ford and Fulkerson algorithm is modified and reinterpreted in order to extract and emphasize its localized information requirements. Very little topological information is needed. Each node needs to know only which nodes are attached to the incoming branches, which are attached to the outgoing ones, and the lengths of the links to the outgoing nodes. A node sends information only to its incoming nodes. For each ultimate destination, a node calculates an estimate of the shortest path via each of its outgoing links; the smallest of these is taken to be its estimate of the shortest path to that destination. All of these approximate shortest distances will have become true shortest distances by the time the algorithm converges. Convergences is guaranteed in finite time, even if the algorithm is implemented by the nodes of the network in an asynchronous manner.

II. THE ALGORITHM

Consider a directed graph consisting of N nodes, denoted $\{1,2,\ldots N\}$, and a collection of arcs, $A = \{(i,j) : i,j \in N$ and there exists a direct link from i to j$\}$. To each arc $(i,j) \in A$ is associated a length $\ell(i,j)$. These lengths could represent physical distance, time, energy, money, or any other quantity suitable for the network of interest. The lengths are unrestricted in sign, but the sum of the lengths in any closed loop of the network is assumed to be positive. Also, for every $i \in N$ define

$$I(i) = \{j : (j,i) \in A\}.$$
$$J(i) = \{j : (i,j) \in A\}.$$

We will refer to $I(i)$ as the set of incoming nodes to i, and $J(i)$ as the set of outgoing nodes from i. Each node maintains a matrix of shortest distances to each ultimate destination via each outgoing branch. In Fig. 1, $d(i,j;k)$ represents the "current shortest distance" from i to j, given that k must be the next node along any path considered,

$$d(i,j) = \min_{k} d(i,j;k),$$

$n(i,j)$ is the next node on the path that achieves the distance $d(i,j)$. Row i is crossed out because it would represent distances from i to itself. For any $j \notin J(i)$, $j \neq i$, column j is crossed out because no direct link exists from i to j.

Initialization of the matrices requires only local topological information. Each node i begins by crossing out row i and all appropriate columns, as discussed previously. The diagonal elements represent direct link distances to other nodes. Thus, every diagonal element which is not crossed out (viz., those in the columns of the outgoing nodes) is assigned the length of the arc from i to the given outgoing node. Assume $\ell(i,j) = M$, a very large number, whenever $(i,j) \notin A$. Then in column i, $d(i,j) = \ell(i,j)$ and $n(i,j) = j$. In other words, since direct paths are the only ones known at this time, column i, which consists of the shortest paths to each destination based on information received to date, initially contains the lengths of the direct links to each node. Note that whenever $d(i,j) = M$, this indicates that no real path from i to j has yet been found. (M should be treated like ∞. $M + d = M$ for any "real distance" d.)

Now, for purposes of analysis, imagine stacking the $N \times N$ distance matrices in order, one above the other, to form a distance cube with the matrix of node 1 at the top, and that of node N at the bottom. The basic idea behind the algorithm is the following. Suppose that node i makes a change (this includes the initialization step) in some $d(i,j)$ component in column i. The only distances that are directly affected by this change are the distances to j, via i, for each of the incoming nodes to i. The distance cube is arranged in such a way that these affected elements are those that are not crossed out and lie along the vertical line that passes through $d(i,j)$. That is, information transmission is purely vertical. Thus, distances to other nodes are received from the set of outgoing nodes, making it possible to calculate distances via these nodes.

More specifically, at each node i the algorithm is begun by initializing the distance matrix. Each distance in column i must then be transmitted up and down the corresponding vertical line. Node i now does nothing until a new distance, say to node k, is received from an outgoing neighbor j. To calculate the distance to k via j, the distance received from j must be added to the direct distance, $\ell(i,j)$, from i to j, which can always be found in the (j,j) element of matrix i. This sum is the new $d(i,k;j)$ and replaces that stored in the (k,j) element of node i's matrix. If it is larger than $d(i,k)$, node i does nothing because a better path has already been found. If it equals $d(i,k)$, j can be included in $n(i,k)$ because there is a tie for shortest path. If it is less than $d(i,k)$, it becomes the "current shortest distance" from i to k, so in the (k,i) element of this distance matrix, node i replaces $d(i,k)$ by $d(i,k;j)$, and $n(i,k)$ by j, and transmits this new distance along the vertical line through his (k,i)

element. The algorithm continues in this manner until no more changes can be made. At this point, each node will know the shortest distance to each destination (or that no path exists, which will be reflected as a shortest path length M), the next node in the path that achieves this distance, and the shortest distance via each alternative outgoing node.

Observe that in any given node's distance matrix the operations in any row are self-contained and independent of those in any other row: the initialization step and each subsequent addition and comparison operation involves only elements in a specific row. This means that the operations performed by a given node for one ultimate destination are independent of those for any other ultimate destination. It has also been noted that in the distance cube constructed by vertically stacking the individual distance matrices, the only communication takes place along vertical lines. This reflects the fact that the information transfer concerning one particular ultimate destination is independent of that for any other. Together, these observations mean that both information transfer and addition-comparisons are independent from one ultimate destination to the next. In the distance cube, this means that the vertical "slices" corresponding to each fixed ultimate destination are self-contained in so far as both communication and addition-comparison operations are concerned. This decomposition property is the basis for our proof of convergence in the next section. It should be emphasized, however, that the topological information required to construct and update each of these vertical slices is not localized. Indeed, interpretation of the algorithm for these vertical planes is closely related to the "centralized" Ford-Fulkerson algorithm. What is important for our purposes is that while the decentralized nature of the algorithm is exhibited by separating the distance cube into horizontal slices, convergence is most easily proven by thinking of the cube as being separated into vertical slices that are self-contained and for which convergence can be proven separately and individually. Clearly, the two are equivalent since they are simply alterative decompositions of the same distance cube.

III. CONVERGENCE OF THE ALGORITHM

The convergence of the algorithm will be proved, by induction, for an arbitrary vertical slice. Since the algorithm can be applied independently to each vertical matrix, convergence for a vertical slice implies convergence for the entire distance cube. Consider the vertical matrix composed of all of the row j's,i.e. corresponding to node j being the common ultimate destination. This matrix will take the form given in Fig. 2. Row j is crossed out because node j is not interested in distances to itself. As an example of the fact that arcs may not exist between all pairs of nodes, the (1,2) element has been crossed out, indicating that in this case the graph contains no direct link from node 1 to node 2. Each diagonal element contains the current shortest distance to destination j and the next node in the corresponding shortest path. Distance changes in any diagonal element will be communicated throughout the corresponding column.

Suppose j is an isolated node, so that no node has a path into j. Then each diagonal element will initially have $d(i,j) = M$. After these M's have all been transmitted, all distances in the vertical matrix will have the value M, no further changes can be made and the algorithm stops. Thus, the final matrix does indeed indicate that no node can find a path to j. Now suppose that at least one node has a path to j. Define

$$S(m) = \{i : \exists \text{ a shortest path from } i \text{ to } j \text{ containing exactly } m \text{ arcs}\}$$

Since for some node at least one path exists to j, an optimal path exists. By the Principle of Optimality, the last link in this path, say (k,j), must be a shortest path from k to j. Thus $k \in S(1)$ and S(1) is not empty. The diagonal element corresponding to each node in S(1) will initially contain the direct distance to j. By the definition of S(1), these distances are optimal. At some time, T(1), each of these distances will have been transmitted throughout its column, and each of the columns corresponding to the nodes in S(1) will now be optimal. In general, define T(m) to be the time at which each node in S(m) has transmitted its true shortest distance to j. Now assume that at time T(m), the columns associated with the nodes in $S(1) \cup S(2) \cup \ldots \cup S(m)$ have been optimized. By the Principle of Optimality, for each node in S(m+1), the shortest paths to j with m+1 arcs must all involve going first through a node in S(m). Therefore, at time T(m), after each of the nodes in S(m) has transmitted its shortest distance, each row corresponding to a node in S(m+1) will have the shortest distance to j in one of its S(m) columns. Since the distances in each of these rows are actual distances, they must be greater than or equal to the shortest distances to j. Thus, after comparisons are performed, the diagonal elements in the S(m+1) rows will contain true shortest distances to j. These distances will be transmitted, and at time T(m+1), the columns corresponding to the $S(1) \cup S(2) \cup \ldots \cup S(m+1)$ nodes will all be optimal. The only fact still to be proven, is that the algorithm stops. Let m' be the smallest integer such that $S(1) \cup S(2) \cup \ldots \cup S(m')$ contains all of the nodes in the graph. The existence of such an integer is guaranteed by the fact that m' cannot exceed N-1, which is the maximum number of links possible in any shortest path. Obviously, at time T(m'), every column in the matrix will consist entirely of optimal distances. At this point, the algorithm must stop.

In order to calculate an upper bound on the number of operations required, the algorithm is modified to operate on a synchronous basis. The event-driven nature of the original algorithm, i.e. sending new distances as soon as they are calculated, is quite convenient for the nodes using the algorithm. However, this characteristic makes it difficult, if not impossible, to bound the number of operations. So we assume instead that nodes are forced to take turns, being allowed to transmit once every N units of time. Define a triple operation to be the sequence of performing an addition, a comparison, and a replacement if necessary. Again, consider an arbitrary vertical slice of the cube. During the first cycle of transmissions, the distances in each of the N-1 diagonal elements will be propagated, and each will cause at most N-2 triple operations. At the conclusion of this cycle, at least one column is optimal, and does not enter into future calculations. During the second cycle, at most N-2 diagonal elements will transmit changes, so a maximum of (N-2)(N-2) triple operations are performed during this cycle. The upper bound for total operations for one vertical matrix is given by

$$(N-2)(N-1) + (N-2)(N-2) + \ldots + (N-2) = \frac{1}{2} N(N-1)(N-2)$$

Therefore, the entire distance cube will be optimal after at most $\frac{1}{2} N^2(N-1)(N-2)$ triple operations. This is an order of magnitude larger than the algorithm in Hu[1], but there are other considerations. The calculations are shared by all of the nodes of the network, and each node does approximately the same amount of work as a central controller would. The principal advantage of the algorithm is that each node computes its own shortest distance matrix.

IV. MODIFICATIONS AND EXTENSIONS

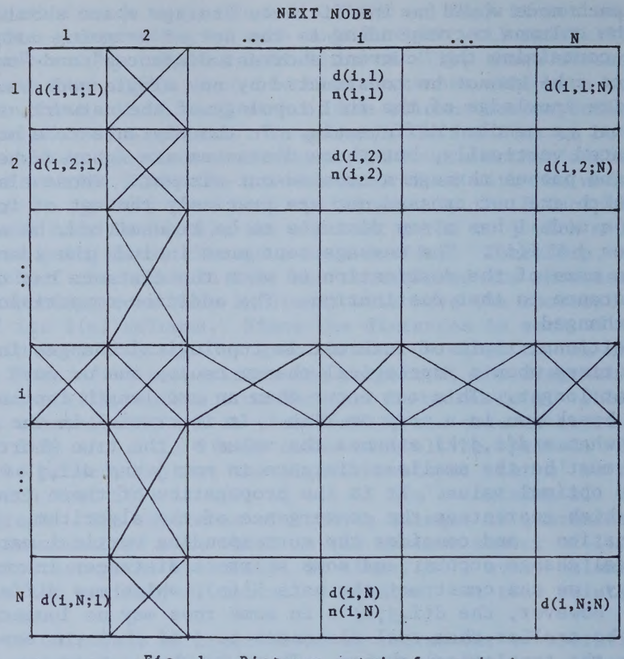
Practical implementation of the algorithm will obviously differ somewhat from the form given above, which has been chosen with ease of presentation in mind. Construction and maintenance of the entire N X N distance matrix by each node would be inefficient. Storage space should be allocated only for the columns corresponding to the set of outgoing nodes, as well as the column containing the "current shortest distances" and "next nodes". The distance cube cannot be constructed by any single node because that would require knowledge of the full topology of the network, so information transfer must be handled differently. In the distance cube model distances are propagated vertically, but these distances are ignored whenever the vertical line passes through a crossed-out element. Those elements along the line which are not crossed out are precisely the set of incoming nodes. Thus, when a node i has a new distance to be transmitted, he sends it only to the nodes $j \in I(i)$. The message sent must include the identity of the sender, the name of the destination to whom the distance has changed, and the new distance to that destination. The addition-comparision operation remains unchanged.

An additional topic of interest is topological changes in the network. A problem arises when a topological change causes one or more shortest paths to get longer. This can occur when an arc length increases or when there is a breakdown in a node or link. It is crucial in the convergence proof that when a d(i,j;k) assumes the value of the true shortest path from i to j, it must be the smallest distance in row j and d(i,j) will be assigned this optimal value. It is the propagation of these true shortest distances which guarantees the convergence of the algorithm. Fix an ultimate destination j and consider the corresponding vertical matrix. Suppose a topological change occurs, and some shortest distances increase. For the new topology, we can construct the sets S'(m), which may differ from the sets S(m). However, the d(i,j;k)'s in some rows may no longer be valid; they could be smaller than real distances to j if they correspond to paths affected by the topological change. Thus, in the convergence proof, there is no guarantee that when a true shortest distance enters a row, it will be smaller than the other distances in the row. It is possible that the algorithm will converge in some such cases, but a result along these lines has not yet been proved.

On the other hand, if a topological change decreases some shortest paths, while none increase, convergence can be proven. Suppose the change occurs, and consider vertical matrix j. Define the sets S'(m) for the new topology. Since the nodes directly affected by a topological change will know about this change immediately, the distances in column j, i.e. the lengths of the direct links into j, will be correct immediately after the topological change. In particular, these direct distances will be in the rows of the S'(1) nodes. The other distances in each of these rows will either be actual distances, or will be larger than actual distances. Therefore, the distance in column j will be the minimum in each S'(1) row, and will be assigned to the diagonal element of the row. Then at time T'(1), the S'(1) columns will be optimal. The same argument is valid in the inductive step of the convergence proof.

V. CONCLUSION

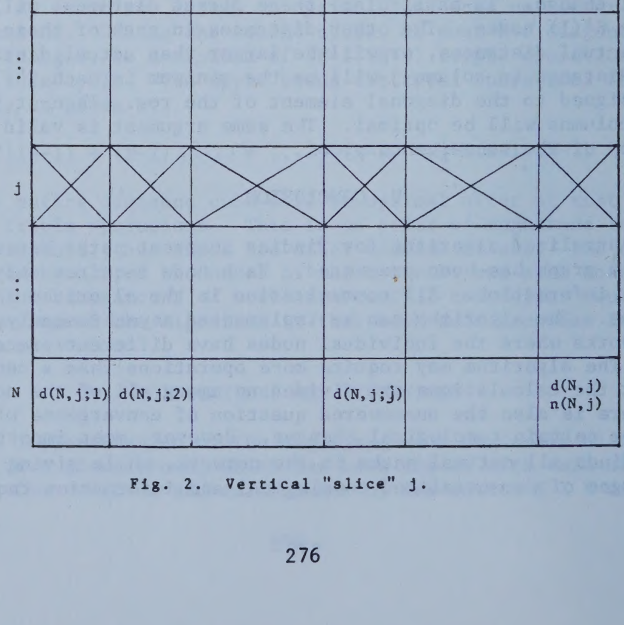
A decentralized algorithm for finding shortest paths between all pairs of nodes in a graph has been presented. Each node requires only local topological information. All communication in the algorithm is between adjacent nodes. The algorithm can be implemented asynchronously, an advanage in networks where the individual nodes have different processing capabilities. The algorithm may require more operations than a centralized scheme, but the calculations are divided up among all of the nodes in the graph. There is also the unanswered question of convergence of the algorithm after certain topological changes. However, most importantly, the algorithm finds all optimal paths in the network, while giving the users the advantages of decentralized topological and information requirements.



NEXT NODE

DESTINATION	1	2	...	i	...	N
1	d(i,1;1)			d(i,1) n(i,1)		d(i,1;N)
2	d(i,2;1)			d(i,2) n(i,2)		d(i,2;N)
⋮						
i						
⋮						
N	d(i,N;1)			d(i,N) n(i,N)		d(i,N;N)

Fig. 1. Distance matrix for node i.



NEXT NODE

NODE OF ORIGIN	1	2	...	j	...	N
1	d(1,j) n(1,j)			d(1,j;j)		d(1,j;N)
2	d(2,j;1)	d(2,j) n(2,j)		d(2,j;j)		d(2,j;N)
⋮						
j						
⋮						
N	d(N,j;1)	d(N,j;2)		d(N,j;j)		d(N,j) n(N,j)

Fig. 2. Vertical "slice" j.

REFERENCES

[1] T. C. Hu, Integer Programming and Network Flows. Reading, Mass.: Addison-Wesley, 1969, pp. 151-161.

[2] S. E. Dreyfus, "An Appraisal of Some Shortest Path Algorithms," Operations Research, Vol. 17, pp. 395-412, 1969.

[3] R. W. Floyd, "Algorithm 97, Shortest Path," Commun. ACM, Vol. 5, p. 345, 1962.

[4] L. R. Ford, Jr., and D. R. Fulkerson, Flows in Networks. Princeton, N.J.: Princeton Univ. Press, 1962, pp. 130-133.

[5] R. Lau, R. C. M. Persiano, and P. P. Varaiya, "Decentralized Information and Control: A Network Flow Example," IEEE Trans. Automat. Contr., Vol. AC-17, pp. 466-473, Aug. 1972.

ACKNOWLEDGMENT

This research was supported by the Office of Naval Research under ONR Contract N00014-76-C-0667.

RECURSIVE EVALUATION OF OVERALL NETWORK RELIABILITY

TALAAT S. EL-SHEIKH,
Dept. Electrical Engineering,
University of Saskatchewan,
Saskatoon, Sask., Canada.

GRANT A. CHESTON,
Dept. Computational Science,
University of Saskatchewan,
Saskatoon, Sask., Canada.

ABSTRACT

In the design of a computer network, one of the most important considerations is the overall reliability of the network. The objective of this paper is to present a technique for calculating the expected number of communicating node pairs and the expected number of nodes that can communicate with a certain node. These calculations are to be done for arbitrarily complex networks when their nodes and arcs fail independently with known probabilities. The general approach is to decompose a complex network into simpler networks until the resulting networks are simple enough to be analyzed using efficient known techniques.

Equations are derived that provide a means for independently calculating the reliabilities of its subnetworks that share only one common node, that is its biconnected components. This enhances the decomposition algorithm to the point that it should have a reasonable execution time for practical networks.

INTRODUCTION

With the state of current technology, both computers and communication links have nontrivial downtimes. If a computer network is to have any degree of reliability, it is necessary to provide for these failures by having alternate paths of communication. Thus the design of the optimal computer network to meet an appropriate reliability criterion is very important.

One of the most important requirements in any network design project is the ability to evaluate the reliability of a specified network. This reliability strongly depends on both the reliability of the individual computer systems and communication links, and the topological layout of the communication links. In this paper, techniques are given for evaluation of computer network reliability for two reliability criteria.

Based on possible definitions of acceptable network operation, many different reliability measures have been defined by many authors [Harary 1962, Frank 1967, Kel'mans 1967, Frank 1969, Hakimi 1969, Boesch 1970, Wilkov 1970, Wilkov 1971, Amin 1973, Kershenbaum 1973, Cavers 1975 and Tainiter 1976]. (See [El-Sheikh 1978] for a discussion of these measures). These measures of network reliability fall into two main classes; those which do not assume failure probabilities on nodes or edges, called deterministic, and those that do, called probabilistic. Since the elements of a computer network are subject to random failure, the probabilistic reliability criteria are more realistic. Hence the present discussion will be limited to probabilistic criteria.

Using the probabilistic criteria, a failure probability is associated with every node and link in the network. Consideration of dynamic failure probabilities is very complicated, so that only static probabilities will be used. The static probability of a node or link being operative Q can be approximated by the relation

$$Q = mut/(mdt+mut)$$

where mdt is the mean down time for the node or link, that is mean time spent in failure waiting for completion of repair; and mut is the mean up time, that is mean time spent in the operational state. These probabilities are limiting state probabilities as time approaches infinity.

There are several static measures of the reliability of a network. Of particular interest are the measures of the overall reliability of the network. The probability that all node pairs can communicate is one simple measure. A somewhat more realistic measure is the probability that all operational nodes can communicate. From a design point of view, these provide measures of the overall network reliability. However the user community in general is likely most interested in the expected number of operational communication paths. This attribute is best measured by the expected number of nodes that can communicate with the central node in a centralized network, and the expected number of node pairs that can communicate in a distributed network. These two measures will be called S and T respectively. Thus our objective is to develop a technique to evaluate these two measures of reliability.

To evaluate a specific measure of reliability, there are generally two approaches. The simulation approach obtains the reliability of a network by statistical and sampling techniques (Van Slyke and Frank [1972] and Van Slyke, Frank and Kershenbaum [1975]). While these techniques are generally efficient, they only yield approximate values for the reliability. We will be interested in analytic techniques to calculate the exact value for the reliability. One approach to this problem is to first calculate the reliability of communication between every pair of nodes and use these values to obtain a value for the overall network reliability. For a large network, not only is the number of node pairs large but also the computational time required to calculate each pairwise reliability becomes prohibitive. Thus we will present a technique that does not require the explicit calculation of the pairwise reliabilities.

Thus the objective of this paper is to design efficient algorithms to analyze the overall reliability of computer networks based on the criteria. The general approach is based on a graph model of the computer network. Thus the second section summarizes the relevant definitions and concepts from graph theory. The following section outlines how the decomposition technique is applied to this problem. Basically the reliability of a network is expressed in terms of the reliability of the two networks obtained by (1) deleting a terminal node or communication link, and by (2) setting the probability of failure for the node or link to zero. By weighing the reliabilities of these networks, the reliability of the original network can be obtained.

While this approach works well in theory, in practice the execution time becomes prohibitively large. To expedite this approach, we present a technique that can be used to allow independent evaluation of subnetworks that only share one common node. The last section gives some comments on the efficiency of the algorithm.

GRAPH THEORETIC CONCEPTS AND DEFINITIONS

In a graph model of a computer network, the computer centers in the network are represented by a set N of n nodes ($N = \{n_i\}$), and the communication links are represented by a set E of b edges or branches that join the node pairs. Nodes n_i and n_j are said to be adjacent if there exists an edge between them and this edge is said to be incident to the two nodes. A simple path exists between nodes n_i and n_j if a sequence of distinct adjacent nodes $[n_i, n_k, \ldots, n_r, n_j]$ exists. The length of a path is equal to the number of edges on it. A closed path (cycle) in a graph results if both end nodes of the path are the same node. A set of nodes is said to be connected if there is a path from any one node to any other node.

A subgraph G_1 of a graph G is a graph with all of its nodes and edges contained in G. In a graph G, a maximal connected subgraph is called a component of G. A (minimal) set of nodes in a graph whose removal increases the number of components by one is called a (Prime) node cutset. If a cutset contains only one node, the node is called a cutpoint. The minimum number of nodes in any cutset, is known as the node connectivity of the graph. A maximal subgraph of connectivity of at least two is known as a biconnected component.

A tree is a connected graph in which there is exactly one path between every pair of nodes. A tree is said to be a spanning tree for a connected graph if it is a subgraph of the graph and contains all the nodes of the graph.

APPLICATION OF THE DECOMPOSITION TECHNIQUE

The general approach to the analysis of the reliability of a computer network is that of the decomposition technique. This technique expresses the reliability of a network in terms of the reliabilities of simpler networks. These networks will be simpler in one of two senses. First the network may have fewer nodes and/or links. If a network is reduced to a single node or a tree, then there are established techniques [Kershenbaum 1973] that can be applied to efficiently determine its reliability. Second, a network is simpler if there is less uncertainty about the reliability of the network. This simplification can be achieved by having more nodes and/or links in a fully operational state. Certainly, if a network contains a spanning tree of fully operational nodes and links, then again the reliability of the network can be readily obtained (there is a fully operational path between every pair of nodes).

To see how this technique is applied to a network, consider the analysis of the reliability of a network, Net, with n nodes and b links. As has been already noted, this analysis is readily done if the network is either a tree or it contains a fully operational spanning tree. Otherwise we select a node or link of the network that isn't fully operational, call it element e. Now the reliability of the network, RL, can be expressed as

$$RL = RL_e^f (1-Q(e)) + RL_e^o \, Q(e)$$

where $Q(e)$ is the probability of element e being operative,

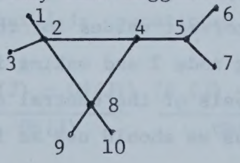
RL_e^f is the reliability of the network, $(Net)_e^f$, obtained by deletion of element e from Net,

RL_e^o is the reliability of the network, $(Net)_e^o$, obtained by changing the reliability of element e in Net to fully operational.

As both $(Net)_e^f$ and $(Net)_e^o$ are simpler networks than the original network, their reliabilities can be evaluated by (recursively) applying the technique to each of them. Eventually, sufficiently simple networks will be obtained that they can be directly analyzed. Then the reliabilities and probabilities of these networks are combined to give the reliability of the original network.

NETWORK SPLITTING AT CUTPOINTS

In 1973, Kershenbaum and Van Slyke [1973] presented analytic techniques for trees and simple cycles to calculate S(J), the expected number of nodes that can communicate with node J, and T, the expected number of communicating node pairs. They also suggested that in a network with cutpoints, such as

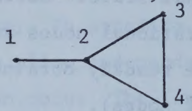


the following approach could be used to find the reliability of the network

- find the reliabilities of the subnetworks {1,2,3}, {4,5,6,7} and {8, 9,10}.
- transfer the reliabilities of the subnetworks to the cutpoints 2, 4 and 8.
- evaluate the central cycle {2,4,8}.

However, this approach is only valid if the central nodes 2, 4, and 8 are all perfect.

To see why this approach does not work, consider the following simple network:

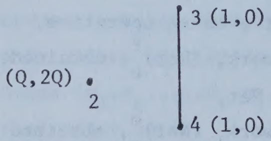


where all nodes and links are assumed to be perfect, except node 2 which has a probability Q of being operative. The analysis of subnetwork {1,2} yields a value of Q for T and a value of 2Q for S at node 2. We now collapse the subnetwork {1,2} to the node 2, transfer the S and T values of the subnetwork to node 2, and proceed to analyze the network {2,3,4} by the decomposition technique. Deleting node 2 and then considering it fully operational, we obtain the following two networks:

(a)

S = 0

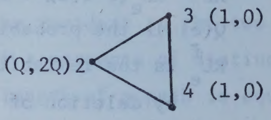
T = 1+Q



(b)

S = 2(1+Q)

T = 1+5Q

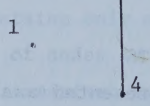


where the values in parenthesis at each node indicate the initial values for S and T respectively. The standard analysis [Kershenbaum and Van Slyke 1973] gives T = 1+Q and S(2) = 0 for network (a), and T = 1+5Q and S = 2(1+Q) for network (b). But networks (a) and (b) are collapsed representations for the networks

(a')

S = 0

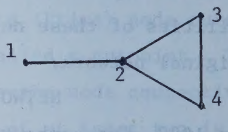
T = 1



(b')

S = 4

T = 6



with their S and T values shown. Clearly transferring the reliabilities of the subnetwork to the cutpoint, and using the standard analysis on the collapsed network yields incorrect values for the reliabilities of the networks derived by deleting node 2 and making it fully operational. The problem is that in the analysis of the central cycle, we consider node 2 as operative or failed. Thus we should use as initial values for node 2, in network (a), the S and T values for subnetwork {1,2} with node 2 failed, and in network (b), the S and T values for subnetwork {1,2} with node 2 operative.

Thus in order to use the approach of Kershenbaum and Van Slyke, we must calculate and store one value for each reliability measure for each state of the cutpoint. Rosenthal [1977] presents a technique that can be used to combine subnetworks with more than one node in common. His approach is an extension of the Kershenbaum and Van Slyke approach in that he calculates and stores one value for each reliability measure for each state of the boundary vertices. Unfortunately, the number of states for the boundary vertices grows superexponentially with the number of boundary vertices. Nevertheless, when there are a small number of boundary vertices and for simple 0-1 reliability measures, this technique can be successfully applied. However, for more complicated measures such as S and T, there are even more states for the boundary vertices.

In this section, we will develop an alternate approach for combining the reliabilities of biconnected components at a cutpoint in order to obtain the reliability of the overall network. The advantage of this approach is that it is only necessary to calculate and store one value for each reliability for each subnetwork and it can be applied to the S and T measures of reliability.

In order to develop the relations required to calculate S and T for a network from the S and T values for subnetworks, consider the network shown in Figure 1. Suppose we have already evaluated the reliabilities of the two subnetworks, and the resulting values are $S_1(J)$ and T_1 for the first subnetwork, and $S_2(J)$ and T_2 for the second subnetwork. We note that both $S_1(J)$ and $S_2(J)$ include the probability that J can communicate with itself, i.e. include QN(J) as an additive term. Thus to avoid adding this term twice, S at J for the overall network is given by

$$S(J) = S_1(J) + S_2(J) - QN(J).$$

To calculate T, we must include the pairs from subnetwork 1, the pairs from subnetwork 2, and the pairs that include one vertex from each subnetwork. The expected number of vertices in a subnetwork that can communicate with J, excluding J itself is given by, S(J) - QN(J). As this value is weighted by the probability that J is operational, (S(J) - QN(J))/QN(J) gives the number assuming that J is operational. Thus the equation for T in the overall network is

$$T = T_1 + T_2 + \frac{(S_1(J) - QN(J))}{QN(J)} \frac{(S_2(J) - QN(J))}{QN(J)} QN(J)$$

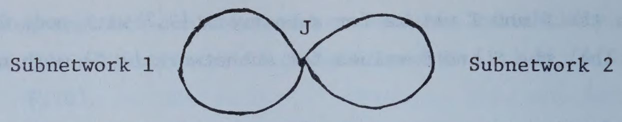


Figure 1: Splitting a network at a cutpoint J.

If a network is composed of m subnetworks all connected at cutpoint J, the same derivation can be used resulting in the following equations:

$$S(J) = \sum_{i=1}^{m} S_i(J) - (m-1)\,QN(J) \tag{1}$$

$$T = T_1 + T_2 + \sum_{i=1}^{m-1} \sum_{k=i+1}^{m} \frac{(S_i(J) - QN(J))\,(S_k(J) - QN(J))}{QN(J)} \tag{2}$$

No complications arise in using this approach to calculate T. However for the S criterion, a complication arises when S must be calculated at some node other than the cutpoint joining the subnetworks (the only node where the equation is applicable). As an example of this, suppose we want to calculate T and S at node 1 for the network shown in Figure 2. We begin by calculating T and S at node 2 for subnetwork 2. Now we can obtain T for the whole network by calculating T and S(2) for subnetwork 1 and then combining the values for the two subnetworks using equation (2). This approach cannot be used for S as it would only yield S(2), whereas we require S(1). When we are calculting S at some node other than the cutpoint, we need to use the modified form of the Kershenbaum and Van Slyke approach.

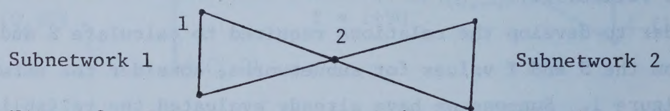


Figure 2: Calculating S at a node other than the cutpoint.

Thus in order to calculate S(1), we collapse subnetwork 2 into node 2 and analyze subnetwork 1 using the appropriate reliabilities for subnetwork 2 to initialize the reliabilities of node 2. When node 2 is failed, the appropriate value is zero. When node 2 is operative, rather than initializing the S value at node 2 to 1, it should be set to

$$\frac{S'(2) - QN(2)}{QN(2)} + 1$$

where S'(2) represents the expected number of nodes in subnetwork 2 that can communicate with node 2. With these values, the analysis of subnetwork 1 can proceed by the standard decomposition approach.

ALGORITHM TIME COMPLEXITY AND FURTHER WORK

While our approach is efficient for combing biconnected subnetworks, the subnetworks must still be evaluated. When the subnetwork has the form of a cycle, the execution time is easily shown to be proportional to the square of the number of nodes (n). As further edges are added to the interior of the cycle, the computational time increases by a factor proportional to n for each added edge. Thus the time increases rapidly with the connectivity of the graph.

In fact, Rosenthal [1977] has shown that just approximating the reliability of arbitrary networks is very difficult for even simple measures. He did this by showing that such problems are NP-Hard so that if a fast algorithm were found for such a problem, it would imply fast algorithms for a whole class of very difficult problems (the NP-Complete class: see Aho, Hopcroft and Ullman [1974] as a general reference). Thus efficient algorithms for the calculation of reliability measures such as S and T cannot be expected. Nevertheless, for networks with low connectivity, the execution time for algorithms such as the one presented here, is not unreasonable. This assumption is not unrealistic for practical networks due to the cost of providing each communication link.

In the above algorithm and analysis, the selection of edges and nodes to be deleted was done arbitrarily. In fact, the selection criterion can have a pronounced effect on the computation time required. As the algorithm already does, cutpoints of the network should be handled first so that the biconnected components are evaluated independently. An implication of this is that the nodes and edges should be removed in an order that leads to cutpoints in the resulting networks. Generally, this ordering is highly dependent on the connectivity of the network. Further analysis of this algorithm is planned towards developing specific rules and/or heuristics for use in the selection criterion. Further work is also planned into extending the approach so that it can be used to combine two subnetworks that share more than one common node.

ACKNOWLEDGEMENT

The authors would like to thank the National Research Council of Canada and the University of Saskatchewan for support while carrying out this research.

REFERENCES

1. Aho, A.V., Hopcroft, J.E. and Ullman, J.D. The Design and Analysis of Computer Algorithms, Addison-Wesley Pub. Com., Don Mills, Ontario, 1974.
2. Amin, A.T. and Hakimi, S.L., "Graphs with given connectivity and independence number or number with given measures of vulnerability and survivability", IEE Transactions on Circuit Theory, Vol. CT-20, No.1, January 1973.
3. Boesch, F.T. and R.T. Thomas, "On graphs of invulnerable communication nets", IEEE Transaction Circuit Theory, Vol. CT-17, pp. 183-191, May 1970.
4. Cavers, J.K., "Cutset manipulations for communication network reliability estimation", IEEE Transaction on Communications, Vol. Com-23, No. 6, pp. 569-575, June 1975.
5. El-Sheikh, T., "Analysis of the reliability of computer networks", Technical Report 78-3, Department of Computer Science, University of Saskatchewan, Saskatoon, Sask., Canada, 1978.
6. Frank, H., "Vulnerability of communication networks" IEEE Trans, Communication Technology, Vol. Com-15, pp. 778-779, Dec. 1967.
7. Frank, H., "Maximally reliable node weighted graphs" in Proc. 3rd Annu. Princeton Con. Information Sciences and Systems, pp. 1-6, Mar. 1969.
8. Hakimi, S.L. "An algorithm for construction of the least vulnerable communication network or the graph with the maximum connectivity", IEEE Transaction on Circuit theory, Vol. CT-16, pp. 229-230, May 1969.
9. Harary, F., "The maximum connectivity of a graph", in Proc. Nat. Acad. Sci., Vol. 48, pp. 1142-1146, July 1962.
10. Harary, F., Graph Theory, Reading, Mass.: Addison-Wesley, 1969.
11. Kel'mans, A.K. "Connectivity of probabilistic networks", Automat. Remote, Contr., no. 3, pp. 444-460, 1957.
12. Kershenbaum, A., Van Slyke, R.M., "Recursive analysis of network reliability", Networks, Vol. 3, pp. 81-94, 1973.
13. Rosenthal, A., "Computing Reliability of Complex Networks", SIAM J. Appl. Math., Vol. 32, No. 2, pp. 384-393, March 1977.
14. Tainiter, M., "A new deterministic network reliability measure", Networks, Vol. 6, pp. 191-204, 1976.
15. Van Slyke, R. and Frank H., "Network Reliability Analysis: Part 1", Networks, 1, pp. 279-290, 1972.
16. Van Slyke, R., Frank, H., and Kershenbaum, A., "Network Reliability Analysis: Part II", Reliability and Fault Tree Analysis, R.E. Barlow, J.B. Fussell and N.D. Singpurwalla, eds., Society for industrial and Applied Mathematics, Philadelphia, pp. 133-152, 1975.
17. Wilkov, R.S., "Construction of maximally reliable communication networks with minimum transmission delay", Proc. 1970 IEEE Int. Conf. Communications, Vol. 6, pp. 4210-4215, June 1970.
18. Wilkov, R.S., "On the design of maximally reliable communication networks", in Proc. 6th Annu. Princeton Conf. Information Sciences and Systems, Mar. 1972.

AN INTERACTIVE CAD PROGRAM FOR CONTROL SYSTEM ANALYSIS AND SYNTHESIS

STANLEY J. LARIMER*
GARY B. LAMONT
Air Force Institute of Technology
Department of Electrical Engineering
Wright-Patterson Air Force Base, OH 45433

ABSTRACT

This paper discusses the development of a computer-aided-design and analysis program for continuous and discrete control systems. The FORTRAN program as described is fully interactive and provides complete error detection and abort protection. In addition to digital and continuous time response, frequency response and root-locus, the program allows block diagram manipulation, state-space analysis and a variety of continuous to digital transformations. Various polynomial, matrix, and scientific calculators are also provided.

Introduction

Control is a field characterized by problems which require substantial computation. Although many of these computations are conceptually simple, they require considerable time and effort to perform. Many computer-aided design (CAD) programs have been written with the intention of reducing this computational effort. (Refs. 2,3,4,5,6,7,8,9,14,22,28,29,34,35 form an incomplete list). Most perform the computations well. Unfortunately many suffer from poorly designed user interfaces which, while not affecting their actual results, have seriously impaired their efficient use. Also, many are not transportable to other computer systems.

Goals

If computer-aided design software is to be improved, it must be designed with two equally important goals in mind: (1) It must be capable of solving whatever class of problems is of interest, and (2) it should require as little effort from the user's standpoint as possible. Only if these goals are accomplished together will users be able to obtain all the benefits of CAD.

The purpose of this paper is to discuss the development of a CAD program which will provide all of these benefits. To this end the following goals were determined:

-To consolidate as many existing CAD guidance and control programs as possible into one package so that users can realize the benefits of a standard input format, the convenience of having every function at their fingertips, and the efficiency of a unified data structure where input does not have to be retyped for every phase of a problem.

-To create new program functions in areas where currently existing tools are unavailable or inadequate.

-To design and develop an efficient user interface which gives the user complete control of the program and its data with a minimum amount of typing.

---

*Now with AF Flight Dynamics Laboratory, WPAFB, OH 45433

-To develop a program structure which is easy to learn, maintain, modify and extend.

Approach

The approach used in this investigation was one of uniform, modular, standardized design. The program is designed "from the bottom up" using an overlay structure for interactive processing and the FORTRAN language for transportability.

The ultimate motivation behind this investigation was the desire to obtain a tool which is capable of performing every computation needed in the field of guidance and control. The following areas were selected as being essential first level elements of a comprehensive computer program for control work:

-Root locus analysis.

-Discrete and continuous time response.

-Discrete and continuous frequency response.

-Block diagram manipulations.

-Polynominal operations.

-Matrix operations.

-Scalar operations.

-Classical control design techniques

-Modern control design techniques.

-Continuous to discrete transformations.

These are the features that are included in the program developed. In addition to being a useful set of functions themselves, they are sufficiently general to allow control development and testing.

The second major goal of this investigation was to develop a program which is not only computationally powerful but also very easy to use. This requires that a user be able to specify, with a minimum amount of input, exactly which operations the program is to perform and what data it is to utilize. The following list defines the features of the program that provide a truly efficient user interface:

-protection against premature program termination due to input errors

-ability to recover from input errors without starting over

-ability to selectively display the value of any variable at any time

-ability to transfer the contents of any one variable to any other variable without manually retyping the data

-ability to provide help to the user at any time, especially when input requested by the program is not understood

-ability to selectively list the options available to the user whenever needed

-ability to stop the program and restart it later without losing any data

-ability to abort a command without terminating the entire program

-ability to assume different modes of operation to tailor fit the program's performance to the user's preferences

-ability to control the flow of the program from function to function with complete freedom and as little effort as possible

Results

A special kind of user assistance program was designed, developed and implemented that teaches a new user how to operate the program, provides quick memory refreshing and prompting for the familiar user and yet is completely invisible to the expert. Specific program options are shown in Table I.

The program to be described as given the name "TOTAL" to reflect the fact that it is intended to eventually perform the total range of computations not only in the control field but in digital signal processing, guidance, navigation and linear systems. Also the name "TOTAL" implies the existance of the interactive capabilities discussed previously. Various hard copy output features are available such as listings, graphs, and plots.

The results of the development include a working computer program, ("TOTAL"), a detailed user's manual, a fully documented programmer's manual, and reference [36]. Many Air Force Institute of Technology students and faculty as well as Wright-Patterson Air Force Base personnel have and are now using "TOTAL" and each has expressed their enthusiasm for its timesaving attributes as well as its generality and completeness.

TOTAL is designed as a tool to be used with as much speed, agility, and confidence as one would use a familiar hand calculator. To help the casual user get a quick idea of TOTAL capabilities, the following overview is provided:

-TOTAL is built around twelve general-purpose polynomials of maximum degree 50, and seven general-purpose 10 x 10 matrices.

-Eight of the polynomials may be paired to form the numerators and denominators of four general-purpose transfer functions.

-With just these polynomials, matrices, and transfer functions, the user is able to use the entire spectrum of TOTAL's capabilities, which include:

- Add, subtract, multiply, divide, factor, expand, and copy polynomials.
- Add, subtract, multiply, invert, transpose, obtain eigenvalues, preset, and copy matrices.
- Obtain root locus, frequency response, and time response in both continuous and digital domain of open and closed-loop transfer functions.
- Perform block diagram reduction.
- Compute transfer functions from state-space.
- Transform between continuous and digital domains using a variety of methods.

-A built-in 20-memory register scientific calculator with a 4-register stack is available to the user at all times.

-The user may quickly list, transfer, or modify any variable, at any time, anywhere in the program.

-Complete error detection, diagnostics, and abnormal termination protection are provided. Specific help is available at any time by simply typing a question mark.

TABLE I

SUMMARY OF TOTAL'S OPTIONS

```
Command- Attach, TOTAL, ID=AFIT

Command- TOTAL

Welcome to Total--Type help for intro--type stop to stop

Option > Help

TOTAL is an interactive computer-aided design program for digital and
continuous control system analysis.  It contains 100 options divided
into groups of 10 according to general application.

Options ending in 0 list the next 10 options, for example, option
30 lists options 30 thru 39.

The following are the main option groups:

  0-9:  Transfer function input options
10-19:  Matrix input options
20-29:  Block diagram manipulation options
30-39:  Time response options
40-49:  Root locus options
50-59:  Frequency response options
60-69:  Polynomial options
70-79:  Matrix operations
```

80-89: Digitization options
90-99: Miscellaneous options

When input is requested by TOTAL, the user may:

enter the requested information, or:
type ? for explanation of input needed,
type L for a list of current variable values,
type * to leave an item unchanged,
type C to use calculator before or during input,
type $ to abort the option,
type X or Y, Z, T, or R1 thru R20 to tell TOTAL to get requested info from corresponding calculator register.

To end TOTAL, type STOP.

Transfer Function Input Options

0 List options
1 Recover all data from file memory
2 Polynomial form--GTF
3 Polynomial form--HTF
4 Polynomial form--OLTF
5 Polynomial form--CLTF
6 Factored form----GTF
7 Factored form----HTF
8 Factored form----OLTF
9 Factored form----CLTF

Matrix Input Options

10 List options
11 AMAT--Continuous system matrix
12 BMAT--Continuous input matrix
13 CMAT--Output matrix
14 DMAT--Direct transmission matrix
15 KMAT--State variable feedback matrix
16 FMAT--Discrete system matrix
17 GMAT--Discrete input matrix
18 Set up state space model of system
19 Explain use of above matrices

Block Diagram Manipulation and State Space Options

20 List options
21 Form OLTF = GTF * HTF (in cascade)
22 Form CLTF = (GAIN*GTF)/ (1 + GAIN*GTF*HTF)
23 Form CLTF = (GAIN*OLTF)/(1 + GAIN*OLTF)
24 Form CLTF = GTF + HTF (in parallel)
25 GTF(s) and HTF(s) from continuous state space model
26 GTF(z) and HTF(z) from discrete state space model
27 Write ADJOINT(sI - AMAT) to file answer
28 Find HTF from CLTF & GTF for CLTF = GTF*HTF/(1+GTF*HTF)
29 Find HTF from CLTF & GTF for CLTF = GTF /(1+GTF*HTF)

Time Response Options

30 List options
31 Tabular listing of F(t) or F(k)
32 Plot F(t) or F(k) at user's terminal
33 Printer plot (written to file answer)

34 Calcomp plot (written to file plot)
35 Print time or difference equation (F(t) or F(k))
36 Partial fraction expansion of CLTF (or OLTF)
37 List T-peak, T-rise, T-setl, T-dup, M-peak, final value
38 Quick sketch at user's terminal
39 Select input: step, ramp, pulse, impulse, sine

Root Locus Options

40 List options
41 General root locus
42 Root locus with a gain of interest
43 Root locus with a zeta (damping) of interest
44 List N points on a branch of interest
45 List all points on a branch of interest
46 List locus roots at a gain of interest
47 List locus roots at a zeta of interest
48 Plot root locus at user's terminal
49 List current values of all root locus variables

Frequency Response Options

50 List options
51 Tabular listing
52 Two cycle scan of magnitude
53 Two cycle scan of phase
54 Plot F(w) at user's terminal
55 Calcomp plot--linear frequency axis
56 Calcomp plot--log frequency axis
57 Tabulate points of interest, peaks, breaks, etc.
58 Calcomp plot--Nyquist polar plot
59 Calcomp plot--Nichols log-mag/angle plot

Polynomial Operations

60 List options
61 Factor polynomial (POLYA)
62 Add polynomials (POLYC = POLYA + POLYB)
63 Sub polynomials (POLYC = POLYA - POLYB)
64 Multiply polys (POLYC = POLYA * POLYB)
65 Divide polys (POLYC + REM = POLYA / POLYB)
66 Store poly_ into Polyd
67 Expand roots into a polynomial
68 (s + a)**^{n} expansion into a polynomial
69 Activate polynomial calculator

Matrix Operations

70 List options
71 ROOTA = eigenvalues of AMAT
72 CMAT = AMAT + BMAT
73 CMAT = AMAT - BMAT
74 CMAT = AMAT * BMAT
75 CMAT = AMAT inverse
76 CMAT = AMAT transposed
77 CMAT = Idenity matrix
78 DMAT = Zero matrix
79 Copy one matrix to another

Digitization Options

80 List options
81 CLTF(s) to CLTF(z) by impulse invariance
82 CLTF(s) to CLTF(z) by first difference approximation
83 CLTF(s) to CLTF(z) by tustin transformation
84 CLTF(z) to CLTF(s) by impulse invariance
85 CLTF(z) to CLTF(s) by inverse first difference
86 CLTF(z) to CLTF(s) by inverse tustin
87 Find FMAT and GMAT from AMAT and BMAT
88 Compute FMAT = exp(AMAT * T)
89 CLTF(X) to CLTF(Y) by X = alpha*(Y + A)/(Y + B)

Miscellaneous Options

90 List options
91 Update memory file with current data
92
93 List current switch settings (echo, answer, etc)
94
95
96 List special commands allowed in option mode
97 List variable name directory
98 List main options of TOTAL
99 Give the introduction to TOTAL

BIBLIOGRAPHY

1. AFIT/EN. *Digital Computer Manual for Faculty and Students of the School of Engineering* (Third Edition). Wright-Patterson Air Force Base, Ohio: Air Force Institute of Technology, July 1975.

2. AFIT/EN. *FREOR. Frequency Response Program.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

3. AFIT/EN. *OPTCON. Solution of the Linear Quadradic Control Problem.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

4. AFIT/EN. *PARTL. Heaviside Partial Fraction Expansion and Time Response Program.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

5. AFIT/EN. *POLY. Polynomial Factoring Program.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

6. AFIT/EN. *ROOTL. User's Manual for a Digital Computer Routine to Calculate the Root Locus.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, August, 1974.

7. AFIT/EN. *SVFB. State Variable Feedback Program.* Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

8. AFIT/EN. ZTRAN. Z-Transform Program to Calculate the Discrete Time Response. Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, 1974.

9. AFFDL-TR-74-69. Digital Flight Control Systems for Tactical Fighters. Honeywell, Inc. Wright-Patterson Air Force Base, Ohio: Air Force Flight Dynamics Laboratory, July 1974.

10. ASD Computer Center. ASD Computer Center Calcomp Plotter Guide (Revision A). Wright-Patterson Air Force Base, Ohio: ASD Computer Center, 1976.

11. ASD Computer Center. ASD Computer Center CDC NOS/BE User's Guide (Revision D). Wright-Patterson Air Force Base, Ohio: ASD Computer Center, 1976.

12. ASD Computer Center. ASD Computer Center Intercom Guide (Revision B). Wright-Patterson Air Force Base, Ohio: ASD Computer Center, 1976.

13. Blakelock, J. H. Automatic Control of Aircraft and Missiles. New York: Joyn Wiley and Sons, Inc., 1975.

14. Cadzow, J. A. and H. R. Martens. Discrete-Time and Computer Control Systems. Englewood Cliffs, N. J.: Prentice Hall, Inc.

15. Control Data Corporation. FORTRAN Extended Version 4 Reference Manual (Revision B, Publication Number 60497800). Sunnyvale, California: Control Data Corporation, 1976.

16. Control Data Corporation. INTERCOM Version 4 Reference Manual (Revision B, Publication Number 60497600). Sunnyvale, California: Control Data Corporation, 1976.

17. Control Data Corporation. Loader Version 1 Reference Manual (Revision J, Publication Number 60342200). Sunnyvale, California: Control Data Corporation, 1976.

18. Control Data Corporation. SCOPE Reference Manual Version 3.4.4. (Revision J, Publication Number 60497600). Sunnyvale, California; Control Data Corporation, August, 1974.

19. Control Data Corporation. UPDATE Reference Manual, (Revision C, Publication Number 60342500). Sunnyvale, California: Control Data Corporation, 1976.

20. D'Azzo, J. J., and C. H. Houpis. Linear Control System Analysis and Design: Conventional and Modern. New York: McGraw-Hill Book Co., 1975.

21. DiStefano III,.S.W. Feedback and Control Systems (Schaum's Outline Series). New York: McGraw-Hill Book Co., 1967.

22. Hornbeck, R. W. Numerical Methods. New York: Quantum Publishers, Inc., 1975.

23. Houpis, C. H., and G. B. Lamont. Short Course on Digital Systems/ Information Processing (Second Edition). Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, July 1976.

24. Kirk, D. E., Optimal Control Theory, An Introduction. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1970.

25. McGillem, C. D., and G. R. Cooper. Continuous and Discrete Signal and System Analysis. New York: Holt, Rinehart, and Winston, Inc., 1974.

26. Meditch, J. S. Stochastic Optimal Linear Estimation and Control. New York: McGraw-Hill Book Co., 1969.

27. Melsa, J. L., and D. G. Schultz. Linear Control Systems. New York: McGraw-Hill Book Co., 1969.

28. Melsa, J. L., and S. K. Jones. Computer Programs for Computational Assistance in the Study of Linear Control Theory (Second Edition). New York: McGraw-Hill Book Co., 1973.

29. O'Brien, F. L. A Consolidated Computer Program for Control System Design. Master's Thesis. Air Force Institute of Technology, December, 1976. ADA 034879

30. Oppenheim, A. V., and R. V. Schafer. Digital Signal Processing. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1975.

31. SofTech, Inc. An Introduction to SADT Structured Analysis and Design Technique. Waltham, Mass.: SofTech, Inc. November, 1976.

32. SofTech, Inc., Structured Analysis Reader Guide. Waltham, Mass.: SofTech, Inc., May, 1975.

33. Whitbeck, R. F., and L. G. Hofmann. Digital Control Law Synthesis in the w' Domain. Hawthorne, California: Systems Technology, Inc., August, 1977.

34. Wiberg, D. M. State Space and Linear Systems (Schaum's Outline Series). New York: McGraw-Hill Book Co., 1967.

35. Young, K. R. The Design of Automatic Control Systems Using Interactive Computer Graphics. Unpublished PhD dissertation. The University of Texas at Austin, May 1974.

36. Larimer, S. J. "An Interactive Computer-Aid-Design Program for Digital and Continuous Control System Analysis and Synthesis", MS Thesis, Air Force Institute of Technology, WPAFB, March 1978. ADA 055418.

AN EFFICIENT INTERACTIVE OPTIMIZATION METHOD FOR MULTI-OBJECTIVE DESIGN PROBLEMS

ANTHONY N. PAYNE AND ELIJAH POLAK
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

We present a man-machine interactive method for multi-objective optimization. The method is specifically designed to cope with the ill-defined, subjective nature of the multi-objective problem and the high cost of computing points in the tradeoff (Pareto optimal) set. With an example, we demonstrate the method's effectiveness as a tool for computer-aided design.

1. INTRODUCTION

We consider the class of multi-objective design problems characterized by a set of feasible alternatives (designs) $X \subset \mathbb{R}^n$ and a multi-objective function $f : \mathbb{R}^n \to \mathbb{R}^m$ which represents m conflicting design objectives. Ideally, the designer wants an $\hat{x} \in X$ such that $f_i(\hat{x}) = \max\{f_i(x) \mid x \in X\}$, $i = 1,2,\ldots,m$. When such an $\hat{x}$ does not exist, the designer must look for an alternative which, in his judgment, gives a reasonable compromise among the m conflicting objectives. Thus, the following tradeoff problem arises: the designer must find a preferred objective value $y^* \in \Gamma(Y)$ and an alternative $x^* \in X$ such that $f(x^*) = y^*$, where

$$Y \triangleq \{y \in \mathbb{R}^m \mid y = f(x),\ x \in X\}$$

is the set of feasible objective values and

$$\Gamma(Y) \triangleq \{y \in Y \mid y' \in Y,\ y \leqq y' \Rightarrow y = y'\}$$

is the set of nondominated (Pareto optimal) objective values (see Figure 1). The fact that the designer's concept of a preferred design is subjective, ill-defined, and often evolving as he explores $\Gamma(Y)$ complicates the solution of this problem.

Many solution approaches to the tradeoff problem have appeared (see, for example, [1]-[3]). A customary, conceptual approach is to construct $\Gamma(Y)$, or an approximation to it [4], from which the designer can estimate y^*. Another approach (as, for instance, in [5], [6]) permits the designer to interactively explore $\Gamma(Y)$ or Y in search of y^*. In both approaches, high computational cost prohibits computing very many points in Y to estimate y^*.

In this paper, we present a man-machine interactive method for multi-objective optimization designed to cope with the subjective nature of the problem and the high cost of computing points in $\Gamma(Y)$. The method is an efficient tool both for approximating $\Gamma(Y)$ when $m = 2$ and for estimating y^* when $m \geq 2$. Since [7], [8] contain the proofs of results in this paper, we omit proofs to shorten our presentation.

Notation. We let $\mathbb{N}(k) = \{1,2,\ldots,k\}$ and $\mathbb{P}(k) = \{N \subset \mathbb{N}(k) \mid N \neq \phi,\ N \neq \mathbb{N}(k)\}$. For sets A and B, $A \sim B = \{a \in A \mid a \notin B\}$. Let $y, y' \in \mathbb{R}^m$. Then $y \leqq y'$ whenever $y_i \leq y'_i$ for all $i \in \mathbb{N}(m)$. We say y' dominates

---

Research sponsored by the Joint Services Electronics Program Contract F44620-76-C-0100, the National Science Foundation Grant ENG73-08214-A01, and the National Science Foundation (RANN) Grant ENV76-04264.

$y(y \leq y')$, if $y \leqq y'$ but $y \neq y'$. We say y' *strictly dominates* $y(y<y')$, if $y_i < y'_i$ for all $i \in \mathbb{N}(m)$. For $b,b' \in \mathbb{R}^m$ with $b \leqq b'$, we call $[b,b'] = \{y \in \mathbb{R}^m | b \leq y \leq b'\}$ an *$\mathbb{R}^m$-interval* or, simply, an interval. The content of an $\mathbb{R}^m$-interval $[b,b']$ is $a([b,b']) = (b'_1-b_1)(b'_2-b_2)\ldots(b'_m-b_m)$ and is a generalization of the concepts of area and volume. ¤

2. INTERVAL REPRESENTATIONS OF TRADEOFF SETS

The set $\Gamma(Y)$ represents the relevant tradeoffs among the m objectives, and we shall refer to it as the *tradeoff set* associated with Y[†]. In many problems, the constraint $x \in X$ imposes a maximum *achievable level*[*] $\bar{c}_i < \infty$ on each objective f_i and, in addition, the designer can specify a *minimum acceptable level* $\underline{c}_i < \bar{c}_i$ for each objective f_i. Therefore, without loss of generality, we shall assume that

$$\Gamma(Y) \subset R_0 \triangleq [\underline{c},\bar{c}].$$

In the sequel, we shall be interested in the class of *all* tradeoff sets which can be drawn in R_0.

Definition 2.1: A nonempty set $\Gamma \subset R_0$ is said to be a *tradeoff set* in R_0 if $y,y' \in \Gamma$ and $y \leqq y'$ imply $y = y'$. The class of all tradeoff sets in R_0 is denoted by $\mathcal{G}(R_0)$. ¤

For any $\Gamma \in \mathcal{G}(R_0)$, we denote by $\mathcal{E}(\Gamma,R_0)$ the set of all points in R_0 which neither strictly dominate nor are strictly dominated by any point in Γ; that is,

$$\mathcal{E}(\Gamma,R_0) \triangleq \{y \in R_0 | \{y' \in R_0 | y < y' \text{ or } y' < y\} \cap \Gamma = \phi\}.$$

Note that $\Gamma \subset \mathcal{E}(\Gamma,R_0)$. Figure 2 depicts a typical Γ and the corresponding $\mathcal{E}(\Gamma,R_0)$ for the $m = 2$ case.

Consider now a $\Gamma \in \mathcal{G}(R_0)$ and a $y \in \mathcal{E}(\Gamma,R_0)$, as illustrated in Figure 3. Then, no points in Γ can be in $\{y' \in R_0 | y' \leq y\}$, the set of points in R_0 which are dominated by y, or in $\{y' \in R_0 | y \leq y'\}$, the set points in R_0 which dominate y. Thus, Γ is contained in the subset of R_0 obtained from R_0 by eliminating these two sets. For $m = 2$, the subset of R_0 containing Γ consists of two rectangles, as shown in Figure 3. For $m \geq 2$, the point y partitions R_0 into a set of 2^m intervals, $\{I(y,T) | T \subset \mathbb{N}(m)\}$, where

$$I(y,T) \triangleq \{y' \in [\underline{c},\bar{c}] | y'_i \in [\underline{c}_i,y_i] \text{ for } i \in T,\ y'_i \in [y_i,\bar{c}_i] \text{ for } i \in \mathbb{N}(m) \sim T\}.$$

Since $I(y,\mathbb{N}(m)) = [\underline{c},y]$ and $I(y,\phi) = [y,\bar{c}]$, the region of R_0 that contains Γ is the union of 2^m-2 of these 2^m intervals (see Figure 3).

Lemma 2.1: Let $\Gamma \in \mathcal{G}(R_0)$. If $y \in \mathcal{E}(\Gamma,R_0)$, then

$$\Gamma \subset R_0 \sim \{y' \in R_0 | y < y' \text{ or } y' < y\} = \bigcup_{T \in \mathbb{P}(m)} I(y,T). \quad ¤$$

[†] For the bi-objective case ($m=2$), $\Gamma(Y)$ corresponds to the familiar "tradeoff curve" between f_1 and f_2.

[*] That is, $\bar{c}_i \triangleq \sup\{f_i(x) | x \in X\} < \infty$ for all $i \in \mathbb{N}(m)$.

Any finite set $E \subset \mathcal{E}_\ell(\Gamma,R_0)$ defines a subregion of R_0 which encloses Γ and which can be expressed as the union of intervals.

Proposition 2.1: Let $\Gamma \in \mathcal{G}(R_0)$. If E is a finite set of points in $\mathcal{E}_\ell(\Gamma,R_0)$, then

$$\Gamma \subset R_0 \sim \bigcup_{y\in E} \{y' \in R_0 \mid y < y' \text{ or } y' < y\}$$

$$= \bigcup_{\{T(y)\mid y\in E\}\subset \mathbb{P}(m)} \; \bigcap_{y\in E} I(y,T(y)).$$ ¤

Note that $\bigcap_{y\in E} I(y,T(y))$ is an interval since it is the intersection of intervals, but it may be empty or of zero content depending on the relative locations of points in E.

For the bi-objective case, we obtain a more explicit form of Proposition 2.1 [7].

Corollary 2.1: For m = 2 and $\Gamma \in \mathcal{G}(R_0)$, if $\{y^i \mid i \in \mathbb{N}(K)\} \subset \mathcal{E}_\ell(\Gamma,R_0)$ with $y_1^i \le y_1^{i+1}$ and $y_2^i \ge y_2^{i+1}$ for all $i \in \mathbb{N}(K)$, then

$$\Gamma \subset R_0 \sim \bigcup_{i\in\mathbb{N}(K)} \{y \in R_0 \mid y^i < y \text{ or } y < y^i\}$$

$$= \bigcup_{i=0}^{K} [y_1^i, y_1^{i+1}] \times [y_2^{i+1}, y_2^i],$$

where $y^0 = (\underline{c}_1, \bar{c}_2)$, $y^{K+1} = (\bar{c}_1, \underline{c}_2)$. ¤

In this case, the subregion of R_0 which encloses Γ consists of K+1 rectangles. Figure 4 illustrates such a situation.

Definition 2.2: Let $\Gamma \in \mathcal{G}(R_0)$. A finite set $E \subset \mathcal{E}_\ell(\Gamma,R_0)$ is said to be a set of experiments for Γ. The set of intervals

$$\{\bigcap_{y\in E} I(y,T(y)) \mid \{T(y) \mid y \in E\} \subset \mathbb{P}(m)\}$$

is said to give an interval representation of Γ. ¤

Figure 4 and Proposition 2.1 suggest that an interval representation of a tradeoff set Γ gives an approximate representation of Γ: the union of the intervals $\bigcap_{y\in E} I(y,T(y))$ encloses and mimics qualitatively the tradeoff set. If each interval has sufficiently small content relative to R_0, then the representation is good. The dimensions of each interval of nonzero content give a fuzzy, qualitative characterization of the tradeoffs possible in the subset of Γ that is contained in the interval. For example, if $[\underline{c}',\bar{c}']$ is an interval in an interval representation of Γ and if $(\bar{c}_i'-\underline{c}_i')$ is "small" but $(\bar{c}_j'-\underline{c}_j')$ is "large", then, in this interval, it may be possible to trade only a "small" decrease in objective i in order to obtain a "large" increase in objective j.

3. DECISION-MAKING MODEL

Motivated by the concept of an interval representation of tradeoff sets, we now propose a procedure the designer can use to systematically explore $\Gamma(Y)$ for an estimate of his preferred point y^*. We assume the designer initially has no information with which to distinguish his actual tradeoff set $\Gamma(Y)$ from any other $\Gamma \in \mathcal{G}(R_0)$. We allow him to observe sets of experiments for $\Gamma(Y)$ to gain additional information about $\Gamma(Y)$.

In a computer-aided design setting, this information is best displayed on a graphics display terminal. For $m = 2$, the interval representation may be displayed directly, as in Figure 4. Unfortunately, for $m \geq 3$ there seems to be no equally satisfactory way of displaying the interval representation. A suggested approach is to display an objective value profile, which consists of a scale for each objective and the components of each experiment plotted on the corresponding scales and connected by a piecewise linear graph, as shown in Figure 5. A set of K experiments partitions each scale into K+1 $\mathbb{R}$-intervals. Then each $\mathbb{R}^m$-interval in the interval representation is a Cartesian product of these $\mathbb{R}$-intervals. A Cartesian product of $\mathbb{R}$-intervals which has elements that either dominate or are dominated by an experiment is not an interval in the representation. Thus an interval $[\underline{c}', \bar{c}']$ is in the representation if, and only if, the piecewise linear graph G connecting the components of an arbitrary point in $[\underline{c}', \bar{c}']$ intersects, at least once, each of the graphs connecting components of an experiment (see Figure 5).

The interactive decision and search procedure for estimating y^* consists of a sequence of stages in which the computer generates experiments for $\Gamma(Y)$. In stage i, the computer displays to the designer the interval representation of $\Gamma(Y)$ defined by $\bar{E}_i$, the set of all experiments from the first i stages. By Proposition 2.1, one of the intervals in representation, say $R(\bar{E}_i)$, will contain y^*. Since the designer's concept of a preferred design may still be evolving, he may select several intervals that possibly contain y^* and eliminate other intervals in which no point would be acceptable to him as y^*. The computer then deploys the next set of experiments to refine the picture of $\Gamma(Y)$ within the selected intervals. At each stage, as the designer learns more about $\Gamma(Y)$, his preferences should become increasingly better defined. The process stops, say after N stages, when the designer eliminates all intervals except for the interval $R(\bar{E}_N)$ containing y^* and this interval gives him a sufficiently good estimate of y^*.

When using this process for $m \geq 3$, a display of the objective value profile for all experiments in $\bar{E}_i$ at stage i may be confusing to the designer. Thus, we can simplify the display problem by displaying at each stage only those intervals which the designer has not already eliminated from consideration.

To derive optimal strategies for deploying experiments, we model the designer as a decision-maker named DM. Given any set E of experiments for a $\Gamma \in \mathcal{G}(R_0)$, DM selects an interval $R(E) \subset R_0$ (containing his preferred point) such that the correspondence $R(\cdot) : E \mapsto R(E)$ satisfies the next two assumptions.

Assumption 3.1: For any set E of experiments for Γ, there exists a set $\{T(y) \mid y \in E\} \subset \mathbb{P}(m)$ such that $R(E) = \bigcap_{y \in E} I(y, T(y))$. ⌑

Assumption 3.2: For sets E and E' of experiments for Γ, if $E \subset E'$, then $R(E') \subset R(E)$. ⌑

Assumption 3.1 requires R(E) to be an interval in the interval representation of Γ defined by E. Assumption 3.2 forbids DM, upon observing E', to include in R(E') any region of R_0 which he would have eliminated had he only observed $E \subset E'$.

Definition 3.1: A correspondence $R(\cdot) : E \mapsto R(E)$ satisfying Assumptions 3.1 and 3.2 for all $\Gamma \in \mathcal{G}(R_0)$ is said to be a DM response function. We denote the class of all DM response functions by $\mathcal{R}$. ¤

Definition 3.2: Let $\Gamma \in \mathcal{G}(R_0)$ and $R \in \mathcal{R}$. Given a set E of experiments for Γ, R(E) is said to be the interval of uncertainty for E and the content of R(E) denoted by a(R(E)), is said to be the uncertainty. ¤

A set E of experiments for Γ gives DM only enough information about Γ for him to say that his preferred point y* is in R(E). However, both the exact location of y* and the character of Γ within R(E) are uncertain; a(R(E)) is a measure of the uncertainty in the location of y*. If a(R(E)) is small relative to the initial uncertainty

$$A_0 \triangleq a(R_0) = \prod_{i=1}^{m} (\bar{c}_i - \underline{c}_i)$$

then the uncertainty in the location of y* is small.

4. SEARCH STRATEGIES

For designers modeled with a response function $R \in \mathcal{R}$ and for any $\Gamma \in \mathcal{G}(R_0)$, we can state in algorithmic form N stages of a search procedure for estimating preferred objective values.

Algorithm 4.1

```
Data:   Initial interval of uncertainty $R_0$.
Step 1: Set $i = 1$, $\bar{E}_0 = \phi$.
Step 2: Compute a set $E_i$ of experiments for $\Gamma$ such that $E_i \subset R_{i-1}$. Set $\bar{E}_i = \bar{E}_{i-1} \cup E_i$.
Step 3: Solicit the DM response to $\bar{E}_i$ and set $R_i = R(\bar{E}_i)$.
Step 4: If $i = N$ stop; else, set $i = i+1$ and go to step 2.
```
¤

Definition 4.1: Any map Σ defined on $\mathcal{G} \times \mathcal{R}$ by an algorithm of the form of Algorithm 4.1, mapping (Γ, R) into a sequence of sets of experiments for Γ, $\{E_1, E_2, \ldots, E_N\}$, is said to be an N-stage sequential search strategy. For given integers $k_1, k_2, \ldots, k_N$, we let $\mathcal{S}_N(k_1, k_2, \ldots, k_N)$, or more simply $\mathcal{S}_N(k_i)$, denote the class of all N-stage sequential search strategies for which the number of experiments in E_i is k_i. ¤

Since the designer's preferred point y* must be estimated with a limited number of experiments (because of computational cost) and without knowing explicitly his response function R or his tradeoff set Γ, we seek an $\mathcal{S}_N(k_i)$ strategy which efficiently and accurately estimates y* for any $\Gamma \in \mathcal{G}(R_0)$ and any $R \in \mathcal{R}$. For this reason, we use the criterion of worst-case uncertainty for comparing the efficiency of different $\Sigma \in \mathcal{S}_N(k_i)$ in estimating y*. Note that a worst-case analysis of this type is a common practice in evaluating the efficiency of search algorithms (cf.[9]).

Definition 4.2: Let $\Gamma \in \mathcal{G}(R_0)$ and suppose $\Sigma \in \mathcal{S}_N(k_i)$ generates the sequence $\{E_1, E_2, \ldots, E_N\}$ of sets of experiments for Γ. Then

$$A(\Sigma, \Gamma) \triangleq \sup_{R \in \mathcal{R}} a(R(\bigcup_{i=1}^{N} E_i))$$

and the worst-case uncertainty is $\sup\{A(\Sigma,\Gamma) \mid \Gamma \in \mathcal{G}(R_0)\}$. ¤

Note that the worst-case uncertainty depends only upon the search strategy, not upon the particular tradeoff set or DM response function.

Definition 4.3: A search strategy $\hat{\Sigma} \in \mathcal{S}_N(k_i)$ satisfying

$$\sup_{\Gamma \in \mathcal{G}(R_0)} A(\hat{\Sigma},\Gamma) \le \sup_{\Gamma \in \mathcal{G}(R_0)} A(\Sigma,\Gamma) \quad \forall \Sigma \in \mathcal{S}_N(k_i)$$

is said to be an optimal $\mathcal{S}_N(k_i)$ search strategy. The minimum worst-case uncertainty $\hat{A}_N(k_i)$ for $\mathcal{S}_N(k_i)$ strategies is then defined by

$$\hat{A}_N(k_i) \triangleq \inf_{\Sigma \in \mathcal{S}_N(k_i)} \sup_{\Gamma \in \mathcal{G}(R_0)} A(\Sigma,\Gamma). \qquad ¤$$

5. OPTIMAL SEARCH STRATEGIES

Bi-objective Case [7]

For the bi-objective case, the following lemma gives a lower bound on the worst-case uncertainty.

Lemma 5.1: If $m = 2$ and $\bar{\Gamma} \triangleq \{y \in R_0 \mid y = t(\underline{c}_1,\bar{c}_2) + (1-t)(\bar{c}_1,\underline{c}_2),\ t \in [0,1]\}$, then

$$A_0 \prod_{i=1}^{N} \frac{1}{(k_i+1)^2} \le A(\Sigma,\bar{\Gamma}) \le \sup_{\Gamma \in \mathcal{G}(R_0)} A(\Sigma,\Gamma),\ \forall \Sigma \in \mathcal{S}_N(k_i). \qquad ¤$$

A strategy which achieves this lower bound is constructed as follows. Suppose $[\underline{c}',\bar{c}']$ is a typical interval of uncertainty and consider a partition of $[\underline{c}',\bar{c}']$ defined by k parallel lines which have slope $(\bar{c}'_2-\underline{c}'_2)/(\bar{c}'_1-\underline{c}'_1)$ and which cut the diagonal line between $(\underline{c}'_1,\bar{c}'_2)$ and $(\bar{c}'_1,\underline{c}'_2)$ into k+1 segments of equal length. Figure 6 shows such a partition when $k = 3$. For general k, the family of equations

$$h(y;\underline{c}',\bar{c}') \triangleq \frac{y_1-\underline{c}'_1}{\bar{c}'_1-\underline{c}'_1} + \frac{\underline{c}'_2-y_2}{\underline{c}'_2-\bar{c}'_2} = 2j/(k+1) \qquad j = 1,2,\dots,k,$$

defines the partition. For any $\Gamma \in \mathcal{G}(R_0)$, a point of intersection of each of the k lines with $\mathcal{E}(\Gamma,R_0) \cap [\underline{c}',\bar{c}']$ can be used as an experiment. Using this idea we have the following search strategy.

Algorithm 5.1 (Bi-objective Strategy $\hat{\Sigma}_N(k_i)$)

Data: $N,k_1,k_2,\dots,k_N$, $R_0 = [\underline{c},\bar{c}]$.

Step 1: Set $i = 1$, $\underline{c}^0 = \underline{c}$, $\bar{c}^0 = \bar{c}$, and $\bar{E}_0 = \phi$.

Step 2: Compute $E_i = \{y^1,y^2,\dots,y^{k_i}\} \subset \mathcal{E}(\Gamma,R_0) \cap R_{i-1}$ satisfying $h(y^j;\underline{c}^{i-1},\bar{c}^{i-1}) = 2j/(k_i+1)$, for $j = 1,2,\dots,k_i$. Set $\bar{E}_i = \bar{E}_{i-1} \cup E_i$.

Step 3: Solicit the DM response to $\bar{E}_i$ and set $R_i = [\underline{c}^i,\bar{c}^i] \triangleq R(\bar{E}_i)$.

Step 4: If $i = N$, stop; else, set $i = i+1$ and go to step 2. ¤

Figures 7 and 8 illustrate the use of $\hat{\Sigma}_2(2,3)$ and $\hat{\Sigma}_3(1)$, respectively.

Lemma 5.2: If $R_1,R_2,\dots,R_N$ are generated by Algorithm 5.1, then

$$a(R_i) \le \frac{a(R_{i-1})}{(k_i+1)^2} \qquad i = 1,2,\dots,N. \qquad ¤$$

This fact and Lemma 5.1 imply $\hat{\Sigma}_N(k_i)$ is optimal.

Theorem 5.1: The minimum worst-case uncertainty for $\mathcal{S}_N(k_i)$ stategies is

$$\hat{A}_N(k_i) = A_0 \prod_{i=1}^{N} \frac{1}{(k_i+1)^2}$$

and the strategy $\hat{\Sigma}_N(k_i)$, given by Algorithm 5.1, is an optimal $\mathcal{S}_N(k_i)$ strategy. ⌑

Despite its conservative nature, the worst-case uncertainty criterion leads to a strategy which reduces the uncertainty quite rapidly (see Lemma 5.2). Note that $a(R_N) \le \hat{A}_N(k_i)$ and, in practice, $a(R_N)$ may be much less than $\hat{A}_N(k_i)$, since $a(R_N)$ depends upon the given tradeoff set and DM response function. Lemma 5.1 indicates that $\bar{\Gamma}$, the diagonal line segment between $(\underline{c}_1,\bar{c}_2)$ and $(\bar{c}_1,\underline{c}_2)$, is a *worst-case tradeoff set*.

Multi-objective Case ($m \ge 2$)

We consider only the class of $\mathcal{S}_N(1)$ strategies. In the bi-objective case, $\hat{\Sigma}_N(1)$, defined by Algorithm 5.1, prescribes placing the i-th experiment in the intersection of $\mathcal{E}(\Gamma,R_0) \cap R_{i-1}$ with the diagonal line segment between $\underline{c}^{i-1}$ and $\bar{c}^{i-1}$. This strategy also applies when $m \ge 2$.

Algorithm 5.2. (Multi-objective Strategy $\hat{\Sigma}_N(1)$)

Data: Initial interval of uncertainty $R_0 = [\underline{c},\bar{c}]$.

Step 1: Set $i = 1$, $\underline{c}^0 = \underline{c}$, $\bar{c}^0 = \bar{c}$, and $\bar{E}_0 = \phi$.

Step 2: Compute $y^i \in \mathcal{E}(\Gamma,R_0) \cap \{y \in \mathbb{R}^m \mid y = (1-t)\underline{c}^{i-1} + t\bar{c}^{i-1}, t \in [0,1]\}$. Set $\bar{E}_i = \bar{E}_{i-1} \cup \{y^i\}$.

Step 3: Solicit the DM response to $\bar{E}_i$ and set $R_i = [\underline{c}^i,\bar{c}^i] \triangleq R(\bar{E}_i)$.

Step 4: If $i = N$, stop; else, set $i = i+1$ and go to step 2. ⌑

Lemma 5.3: For $m \ge 2$, if $R_1, R_2, \ldots, R_N$ are generated by Algorithm 5.2, then

$$a(R_i) \le \frac{(m-1)^{m-1}}{m^m} a(R_{i-1}) \quad i = 1,2,\ldots,N. \qquad ⌑$$

Theorem 5.2: For $m \ge 2$, the minimum worst-case uncertainty for $\mathcal{S}_N(1)$ strategies is

$$\hat{A}_N(1) = \left(\frac{(m-1)^{m-1}}{m^m}\right)^N A_0$$

and the strategy $\hat{\Sigma}_N(1)$, given by Algorithm 5.2, is an optimal $\mathcal{S}_N(1)$ strategy. ⌑

Computation of Experiments

The optimal search strategies constrain each experiment to be in the intersection of $\mathcal{E}(\Gamma,R_0)$ with certain line segment. An experiment satisfying such a constraint can be characterized by a mathematical program. For example, it is easy to show [8] that the i-th experiment in Algorithm 5.2 can be defined by

$$y^i = (1-\lambda_i)\underline{c}^{i-1} + \lambda_i \bar{c}^{i-1}.$$

where

$$\lambda_i \triangleq \max\{\lambda \,|\, (1-\lambda)\underline{c}^{i-1} + \lambda \bar{c}^{i-1} \leqq y,\ y \in Y,\ \lambda \in [0,1]\}$$

A similar result can be derived for Algorithm 5.1.

6. EXAMPLE

We now illustrate how a designer might use the optimal strategies to look for a preferred design for a two-winding shell-type transformer rated at 1000VA and 60Hz. The mathematical model and material specifications (type 1) are in [10]. The design objectives are (i) minimize the cost (units of money, um) of the transformer and (ii) for high efficiency minimize the total losses. Besides the constraints to guarantee physical realizability and the VA rating, the flux density must be less than 1.75 wb/m^2 to prevent distortion and the cost should not exceed 800 um. For these constraints, the tradeoff set between cost and losses is in the rectangle A_1 shown in Figure 9.

To get a rough picture of the tradeoffs between cost and losses, the designer first uses $\hat{\Sigma}_1(3)$ to place three experiments in A_1 and obtains rectangles B_1, B_2, B_3, B_4 shown in Figure 9. Having this information, he eliminates B_1 since losses in B_1 are unacceptable to him. Also, B_4 does not offer much improvement in losses over B_3, so he eliminates B_4. Thus, the designer is left with B_2 and B_3 to consider. He now applies $\hat{\Sigma}_1(1)$ to B_3 and obtains the rectangles C_1 and C_2. On comparing C_1 and C_2 he prefers C_1 because its points have lower cost than those in C_2 with only a slight increase in losses. He then applies $\hat{\Sigma}_2(1)$ to B_2, deciding to decrease losses after each experiment. Thus, he obtains rectangle D_2. At this point, the designer has ascertained that his preferred point is in C_1 or D_2. He may continue to refine his picture in C_1 and D_2 until he obtains a sufficiently good estimate of his preferred values of cost and losses.

7. CONCLUSIONS

The interactive method present in this paper has several features which make it attractive for the multi-objective design problem. First, it can cope with the ill-defined nature of the problem and affords the designer much flexibility in its use. Second, it makes very simple assumptions about the designer's ability to process information and express preferences. The designer is not required to deal with numerical tradeoff ratics as, for example, in [5], [6], [11]. Third, the man-machine dialogue is ideally done via a graphics display terminal. Finally, there are strategies for selecting experiments which make the method efficient.

8. REFERENCES

[1] B. Roy, "Problems and methods with multiple objective functions," Mathematical Programming, vol. 1, pp. 239-266, Nov. 1971.

[2] R. L. Keeney and H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, New York: John Wiley and Sons, 1976.

[3] E. Polak and A. N. Payne, "On multicriteria optimization," in Directions in Large-Scale Systems, Y. C. Ho and S. K. Mitter, Eds. New York: Plenum, 1976, pp. 77-94.

[4] E. Polak, "On the approximations of solutions to multiple criteria decision making problems," in Multiple Criteria Decision Making: Kyoto 1975, M. Zeleny, Ed. Berlin: Springer-Verlag, 1976, pp. 271-282.
[5] A. M. Geoffrion, J. S. Dyer, and A. Feinberg, "An interactive approach for multi-criterion optimization with an application to the operation of an academic department," Management Science, vol. 19, pp. 357-368, Dec. 1972.
[6] J. S. Dyer, "Interactive goal programming," Management Science, vol. 19, pp. 62-70, Sept. 1972.
[7] A. N. Payne and E. Polak, "An interactive rectangle elimination method for bi-objective decision-making," Electronics Research Lab., Memo UCB/ERL M78/50, Univ. of Calif., Berkeley, May 1978.
[8] A. N. Payne, "Interval elimination methods for interactive optimization of multiple objectives," Ph.D. Thesis, University of California, Berkeley, California, 1978.
[9] J. Kiefer, "Optimum sequential search and approximation methods under minimum regularity assumptions," J. Soc. Indust. Appl. Math., vol. 5, pp. 105-136, Sept. 1957.
[10] H. C. Hamaker and Th. Hehenkamp, "Minimum-cost transformers and chokes," Philips Res. Rep., vol. 5, pp. 357-394, Oct. 1950, vol. 6, pp. 105-134, April 1951.
[11] Y. Y. Haimes and W. A. Hall, "Multiobjectives in water resources systems analysis: the surrogate worth trade-off method," Water Resources Res., vol. 10, pp. 615-624, Aug. 1974.

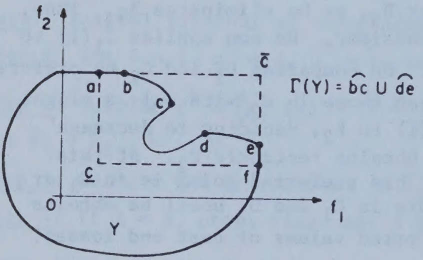

f_2
a
b
$\bar{c}$
$\Gamma(Y) = \widehat{bc} \cup \widehat{de}$
c
d
e
$\underline{c}$
f
0
f_1
Y

Fig. 1. Tradeoff Set $\Gamma(Y)$.

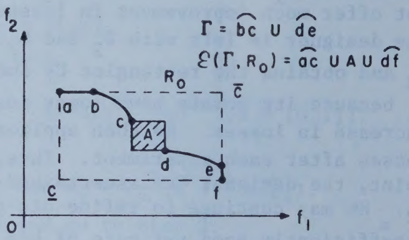

f_2
$\Gamma = \widehat{bc} \cup \widehat{de}$
$\mathcal{E}(\Gamma, R_0) = \widehat{ac} \cup A \cup \widehat{df}$
R_0
$\bar{c}$
a
b
c
A
d
e
f
$\underline{c}$
0
f_1

Fig. 2. Tradeoff Set Γ and $\mathcal{E}(\Gamma,R_0)$

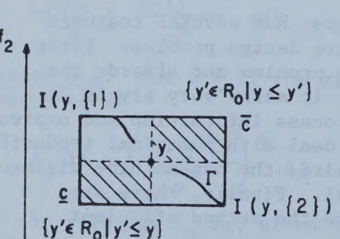

f_2
$I(y,\{1\})$
$\{y' \in R_0 | y \leq y'\}$
$\bar{c}$
y
Γ
$\underline{c}$
$I(y,\{2\})$
$\{y' \in R_0 | y' \leq y\}$
0
f_1

Fig. 3. Subset of R_0 containing Γ.

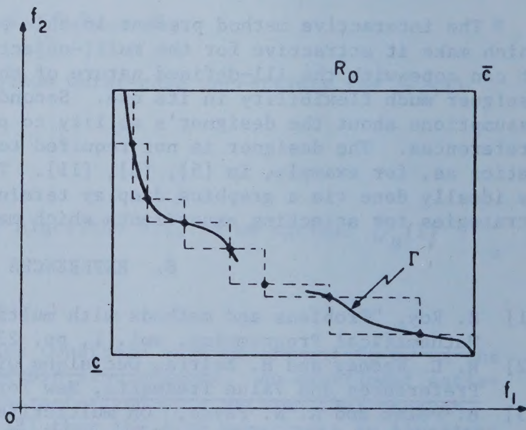

f_2
R_0
$\bar{c}$
Γ
$\underline{c}$
0
f_1

Fig. 4. 7-Experiment Interval Representation of Γ.

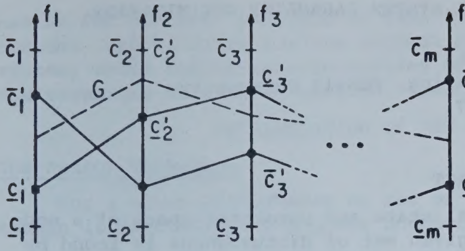

f1
f2
f3
fm
c̄1
c̄1'
c1'
c1
G
c̄2
c̄2'
c2'
c2
c̄3
c3'
c̄3'
c3
c̄m
c̄m
cm'
cm
...

Fig. 5. Objective Function Profile for 2 Experiments.

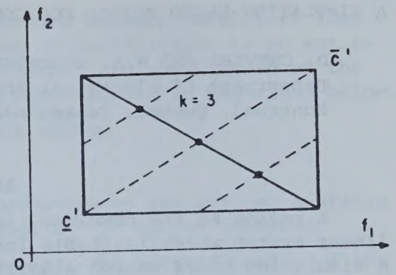


Fig. 6. Partition of $[\underline{c}',\bar{c}']$.

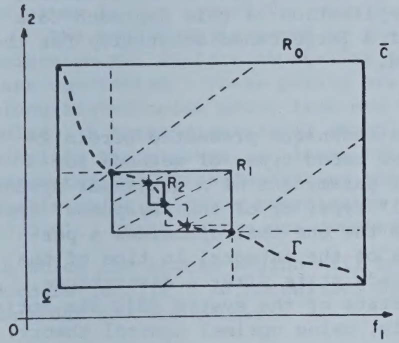

f2
R0
c̄
R1
R2
Γ
c
0
f1

Fig. 7. Strategy $\hat{\Sigma}_2(2,3)$.

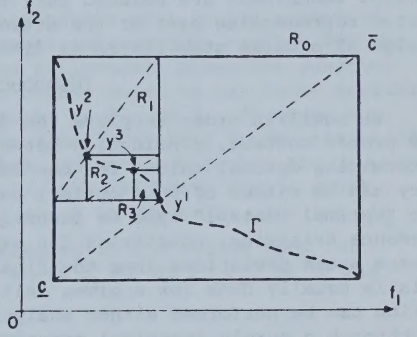

f2
R0
c̄
R1
y2
y3
R2
R3
y1
Γ
c
0
f1

Fig. 8. Strategy $\hat{\Sigma}_3(1)$.

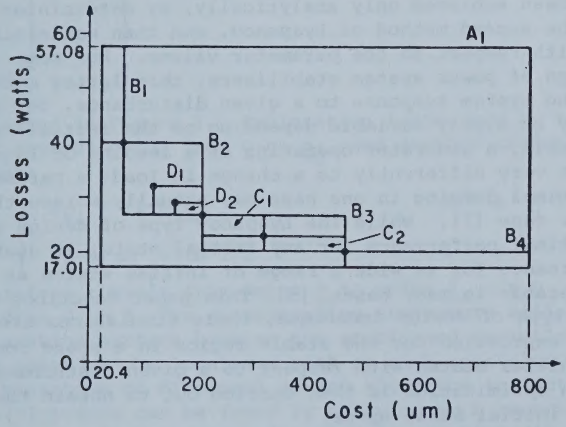


Fig. 9. Cost-Losses Tradeoffs for Transformer Design Example.

A SIMULATION-BASED METHOD FOR CONTROL SYSTEM PARAMETER OPTIMIZATION

D. CREVIER AND M.A. NOURMOUSSAVI
Department of Electrical Engineering, McGill University
Montreal, Quebec, Canada H3A 2A7

ABSTRACT

A region in the combined initial state and parameter space of a nonlinear system which is stable for a given set of disturbances is found by a simulation based search algorithm. An analytical approximation for this stable region is then used to find the parameter value maximizing the stable volume in initial state space.

The advantages of the above method over optimal control or classical control techniques are pointed out; the application of this approach to a system representing part of the network of a large Canadian utility for the design of machine stabilizers is discussed.

INTRODUCTION

We shall, in order to place the design technique presented herein in its proper context, distinguish between two broad types of methods for determining optimal values for the control parameters of a nonlinear system: they can be either of the "optimal control" type, or of the "Lyapunov" type. The "optimal control" type of technique is the one that optimizes a performance criterion, consisting for example of the integral in time of the system state deviations from the final steady state after a disturbance. This is usually done for a given initial state of the system [2]. The optimization can be performed either analytically, using optimal control theory, or through a purely numerical procedure where the performance criterion and its derivatives with respect to parameter values are computed by repeated simulations of the system.

Design techniques of the "Lyapunov" type will try to maximize the region in state space where the system is stable. To the authors knowledge, this has so far been achieved only analytically, by determining a stability region through the second method of Lyapunov, and then maximizing the volume of that region with respect to the parameter values. For some applications, such as the design of power system stabilizers, this latter method is preferable, since the system response to a given disturbance, for given parameter values, may be highly variable depending on the initial state of the system. For example, a generator operating at a leading or lagging power factor will react very differently to a change in load: a parameter value which induces optimal damping in one case may actually worsen the performance in the other case [1]. While the Lyapunov type of design method does not guarantee optimal performance for any initial state, it guarantees acceptable performance for as wide a range of initial states as possible, which seems preferable in many cases [5]. This paper describes a simulation based "Lyapunov" type of design technique, where simulations are used to obtain a general expression for the stable region in a space consisting of parameters and initial states with respect to a given disturbance or set of disturbances. An optimization is then carried out to obtain the largest stable volume in initial state space.

The advantage of such a procedure is that no approximations are necessary on the analytical representation of the system, since all the information on system behavior is extracted from simulations. Systems which would be much too complex for analytical treatment can thus be dealt with in a very simple fashion. In the case of power systems, for example, the optimal control of electromechanical transients has been studied mostly for systems comprising only one generator against an infinite bus [3,4]. It has so far not been possible to take interactions between generators into account, and to solve for the optimal damping of oscillations in an entire system, while taking non-linearities into account. An application of the new technique to a three machine, twentieth order system is described below.

DETERMINATION OF THE STABLE REGION

The Search Technique

For a given disturbance or set of disturbances on the system, a stable domain is identified in a given bounded region Ω_{x_s,x_p} of the combined initial state and parameter space of the system. The identification of the stable domain is performed by a simulation-based search technique which finds points on the stability boundary. An in-depth description of the search has already appeared in reference [6], and only a brief review of it is given in the sequel for completeness.

The technique is based on the determination of a finite number of points on the surface separating the stable and unstable regions for the case considered. These points are found by interval bisection searches along trajectories going from one known stable point to one known unstable point. A three-dimensional example appears in Figure 1. In order to simplify the algebra, a linear transformation is applied to the state and parameter variables describing the system, in such a way that the known unstable point acquires coordinates (1,1,1), and the known stable point, (0,0,0). These points are denoted by $\underline{1}$ and $\underline{0}$ in Figure 1. The set $\Omega_{x_s\ x_p}$, in which the stable region will be determined is the square "box" of corners $(\varepsilon_1 \ldots \varepsilon_n)$ in normalized space, where $\varepsilon_i = 0$ or 1, $\forall i$, as in Fig. 1. Consider now the points:

$$\underline{p}_j = \frac{1}{2}\begin{bmatrix} 1-\alpha \\ 1-\alpha \\ \cdot \\ \cdot \\ 1+\alpha \\ \cdot \\ \cdot \\ \cdot \\ 1-\alpha \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \leftarrow j^{\text{th}}\ \text{entry} \\ \\ \\ \\ \\ \end{matrix} \qquad \begin{matrix} 0 \leq \alpha \leq 1 \\ j = 1 \ldots n \end{matrix} \tag{1}$$

$\underline{p}_1$ is illustrated in Figure 1. Exploration trajectory $\underline{t}_j$ then consists of one straight segment going from $\underline{0}$ to $\underline{p}_j$, and another segment, going from $\underline{p}_j$ to $\underline{1}$. Thus:

$$\underline{t}_j = \begin{cases} b\underline{p}_j + (1-b)\,\underline{0} & 0 \leq b \leq 1 \\ (2-b)\,\underline{p}_j + (b-1)\,\underline{1} & 1 \leq b \leq 2 \end{cases} \tag{2}$$

Point $\underline{t}_j$ therefore travels from point $\underline{0}$ to point $\underline{1}$, via $\underline{p}_j$, as parameter b is varied from 0 to 2. The corresponding trajectory will therefore intersect the separation surface between the stable and unstable regions. The intersection point can be localised with as much precision as desired by an interval bisection of Fibonnaci search along the trajectory. n linearly independent [6] points can be found by carrying such searches on trajectories $\underline{t}_j$, j=1...n, for a given value of α in equation (1). Still more points can be found by repeating the procedure for other values of α.

The Separation Surface

Let $\underline{m}_1, \underline{m}_2 \ldots \underline{m}_n$ denote the points found above on the separation surface. An analytical expression can then be fitted to these points to approximate the separation surface. Linear and quadratic expressions have been tried by the authors as reported in [6]. Better results have been

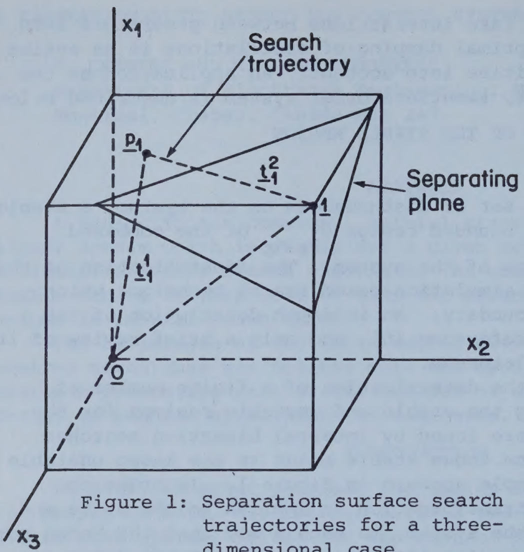


Figure 1: Separation surface search trajectories for a three-dimensional case.

obtained since with the method described below, which has the advantage of not a priori assuming any particular shape for the separation surface. The method is inspired from the technique of potential functions, used in Pattern Recognition [7]. There are however differences, and to avoid confusion it will be referred to in this paper as the Exponential Function Approximation [9].

Let points $\underline{x}_1^s$, $\underline{x}_2^s \ldots \underline{x}_n^s$ denote points lying on the stable side of the separation surface, as follows: $\underline{x}_i^s$ will lie on exploration trajectory i, at a euclidian distance d from point $\underline{m}_i$, measured along the trajectory. Similarly, let $\underline{x}_1^u$, $\underline{x}_2^u \ldots \underline{x}_n^u$ be points on the unstable side, along the exploration trajectories, at a distance d from $\underline{m}_1$, $\underline{m}_2 \ldots \underline{m}_n$. Experimentation demonstrates that d can be chosen to be half the width of the remaining interval in the bisection search along the trajectories [9]. Let now

$$Y_1(\underline{x}) = \sum_{i=1}^{M} a_i \; \mathrm{EXP}\{-\frac{1}{2}(\underline{x} - \underline{x}_i^s)^T \; \underline{\Sigma}^{-1} (\underline{x} - \underline{x}_i^s)\} \tag{3}$$

$$Y_2(\underline{x}) = \sum_{i=1}^{M} \mathrm{EXP}\{-\frac{1}{2}(\underline{x} - \underline{x}_i^u)^T \underline{\Sigma}^{-1} \; (\underline{x} - \underline{x}_i^u)\} \tag{4}$$

where $\underline{\Sigma}$ is the covariance matrix of points $\underline{m}_1$, $\underline{m}_2 \ldots \underline{m}_n$, with off diagonal elements set to zero (as demonstrated in [7], the accuracy of the result is not critically dependent on $\underline{\Sigma}$, and such approximations as neglecting off diagonal terms are acceptable),and M is the total number of points on the separation surface. The a_i coefficients are chosen to satisfy the equations:

$$Y_1(\underline{m}_i) - Y_2(\underline{m}_i) = 0 \qquad i = 1 \ldots n \tag{5}$$

An analytical approximation for the separation surface is then:

$$Y_1(\underline{x}) - Y_2(\underline{x}) = 0 \tag{6}$$

The condition (5) guarantees that this approximation passes through the experimental points $\underline{m}_1$, $\underline{m}_2 \ldots m_n$. A two dimensional representation of a separation surface is displayed in Figure 2.

THE SEPARATION SURFACE AS A GENERAL METHOD FOR OPTIMISATION

Consider the example in Figure 3 : x_s is a state variable, x_p a parameter variable, and the region below the curve (or separation surface) is stable. If there exists an optimal value for the parameter, it can be determined from the separation surface. In Figure 3, this optimal value is x_p^*, the value that allows the maximum range of variation of state variable x_s in the stable region. If there were more state variables, the optimal parameter value could be taken to be the one that would maximize the volume of the stable region in state space.

Determination of stable volume in state space as a function of parameter variables.

The stable volume for a given value of the parameter vector will be

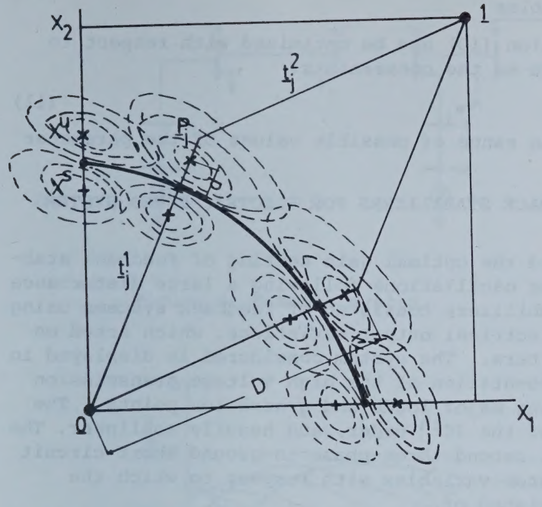


Figure 2: Exponential surface approximation for separation surface.

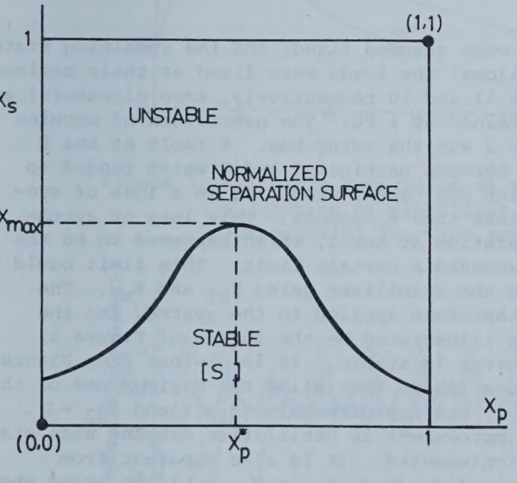


Figure 3: Normalized operating region for a two dimensional case.

the volume in state space of the region [S] bounded by the separation surface, the "box" Ω_{xs}, x_p of Fig. 3, and other constraints on the initial state if such are present (e.g. load flow feasibility constraints in the case of power systems). Let this volume be denoted by $V(\underline{x}_p)$. Then

$$V(\underline{x}_p) = \int_{[s]} dx_{s1}\, dx_{s2} \dots dx_{sn} \qquad (7)$$

The authors have not been able to find an analytical technique for evaluating the above stable volume. Numerical calculation is impossible because a very large amount of computer time might be necessary for large systems. The stable volume can however be estimated by using a Monte-Carlo approach [8].

For a given value of $\underline{x}_p$, random values of the state vector $\underline{x}_s$ are generated in the square "box" Ω_{x_s}, x_p in state space. Let $\underline{x}_s^r$ be such a point. If it lies in region [S] above, it is labeled a success, otherwise a failure. After a number of such points have been generated and labeled, an estimate for the volume is:

$$V = \frac{N_s}{N_r} \qquad (8)$$

where

N_s = No. of Successes
N_r = No. of Random Points

The variance of the error can be shown to be [9];

$$\sigma_V = \sqrt{\frac{V - V^2}{N_r}} \qquad (9)$$

The above procedure may be carried out for several values of the parameter vectors. Then the volume can be approximated by a quadratic equation.

$$V = r + \underline{q}^T \underline{x}_p + \underline{x}_p^T \underline{P}\, \underline{x}_p \qquad (10)$$

The parameters r, $\underline{q}$ and $\underline{P}$ can be obtained by carrying out a least square fit of the above equation to the volumes calculated above.

Optimisation of parameter variables

The function V of equation (10) can be optimized with respect to the parameter vector $\underline{x}_p$, subject to the constraints:

$$0 \le x_{pi} \le 1 \qquad \forall i \tag{11}$$

These inequalities represent the range of possible values of the parameter variables in normalized space.

APPLICATION: DESIGN OF FEEDBACK STABILIZERS FOR ELECTRIC POWER SYSTEMS

The method was used to find the optimal gain setting of feedback stabilizers to control synchronizing oscillations following a large disturbance in a power system [9]. The stabilizers consisted of feedback systems using as input signals the machine electrical output variations, which acted on the machine static voltage exciters. The system considered is displayed in Figure 4: it is a reduced representation of the high voltage transmission system of Hydro-Quebec, including major loads and generation points. The resulting dynamical system is of the 20th order, and heavily nonlinear. The disturbance consisted in a 0.11 second three-phase-to-ground short circuit at bus 8. The state and parameter variables with respect to which the optimization was performed consisted of:

- one state variable: the active generation at machine 1 (PG1) before the fault
- two parameter variables: the stabilizer gains of machines 1 and 2: K_{s1} and K_{s2}.

Other parameter variables were assumed fixed, and the remaining state variables were accounted as follows: the loads were fixed at their maximum values (75 and 37.5 PU at buses 11 and 10 respectively, zero elsewhere) and all voltages were assumed maintained at 1 PU. The generation of machine 3 was fixed at 30 PU, and machine 2 was the swing bus. A fault at bus 8 would then induce oscillations between machines 1 and 2 which tended to build up if no stabilizing action was taken, resulting in a loss of synchronism of machine 1 or 2 in less than 8 seconds. This loss of synchronism occured only if the generation at bus 1, which happened to be the largest unit in the network, exceeded a certain limit. This limit could be increased by properly adjusting the stabilizer gains K_{s1} and K_{s2}. The procedure described above was therefore applied to the system, and the resulting separation surface is illustrated by the curves of Figure 5, where the region "below" the curves is stable. It is obvious from Figure 5 that the optimal parameter values (those that allow the maximum use of the generating capacity of machine 1) are approximately $K^*_{s1} = 1$ and $K^*_{s2} = 0$. Fig. 6 displays the resulting improvement in oscillation damping when stabilizers with these gains are implemented. It is also apparent from Figure 5 that "wrong" parameter values (e.g. $K_{s1} = K_{s2} = 2$) can bring about a behavior of the system worse than when no stabilizers are present ($K_{s1} = K_{s2} = 0$). This effect is also illustrated in Figure 6.

Although the optimal gain values can be determined visually from Fig. 6 in this case, the maximization of the stable volume can be performed for illustrative purposes. The following analytical expression for the volume was obtained:

$$\begin{aligned} V(x_{p1}, x_{p2}) &= PG1(K_{s1}, K_{s2}) \\ &= 0.150 + 0.472\,K_{s1} + 0.409\,K_{s2} - 0.132\,K_{s1}^2 \\ &\quad - 0.307\,K_{s2} - 0.095\,K_{s2}^2 \end{aligned} \tag{12}$$

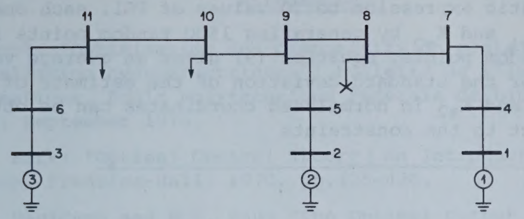


Figure 4: Example system.

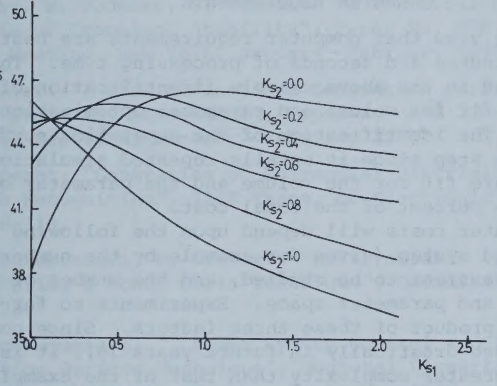


Figure 5: Allowable generation at machine #1 as a function of stabilizer gains K_{s1} and K_{s2}. (Exponential function approximation).

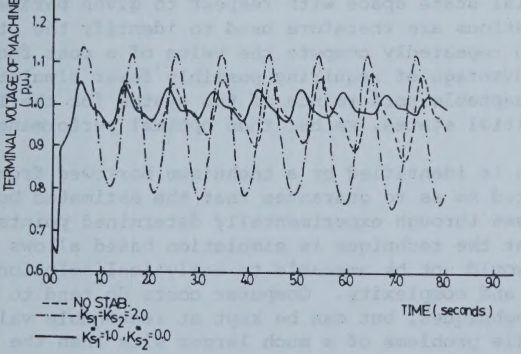


Figure 6: Terminal voltage oscillations at machine 2 following disturbance for different stabilizer gains.

This equation is valid for normalized parameter values, and was obtained by fitting a quadratic expression to 30 values of PG1, each one obtained for given values of K_{s1} and K_{s2} by generating 1500 random points in Ω_{x_s,x_p}. For this number of random points, equation (9) gives an average value of approximately 0.005 for the standard deviation of the estimate of V. The optimal values of K_{s1} and K_{s2} in normalized coordinates can be obtained by minimizing (12) subject to the constraints

$$\begin{aligned} 0 \le K_{s1} \le 1 \\ 0 \le K_{s2} \le 1 \end{aligned} \qquad (13)$$

This can be done by a technique such as Rosenbrock's hill-climbing algorithm [9]. The maximum happens to lie on one of the constraints and corresponds to $K_{s1} = 0.533$, $K_{s2} = 1$. After denormalization, the values become $K^*_{s1} = 0.95$ and $K^*_{s2} = 0$, which agrees with the appearance of Figure 6.

COMPUTER REQUIREMENTS

We shall take the view that computer requirements are best measured in dollars rather than minutes and seconds of processing time. The total computer costs involved in the above example (identification of separation surface, least square fit for volume and parameter optimization) were of the order of $30.00. The identification of the separation surface is by far the most expensive step since it entails repeated simulations of the system; the least square fit for the volume and the parameter optimization account only for a few percent of the total cost.

In general, computer costs will depend upon the following factors: the size of the studied system (given for example by the number of states), the duration of the transient to be studied, and the number of variables included in the state and parameter space. Experiments so far indicate that the costs vary as the product of these three factors. Since computer costs are expected to decrease drastically in future years [6], it is apparent that systems of much greater complexity than that of the example above are within the grasp of the method.

CONCLUSION

A simulation based optimization technique for dynamical systems has been proposed and illustrated. The goal of the method is to maximize the stable region in initial state space with respect to given parameter variables. The simulations are therefore used to identify the stable region, rather that to repeatedly compute the value of a cost functional. This has the double advantage of requiring possibly fewer simulations, and of guaranteeing an acceptable performance of the system for the largest possible number of initial states, rather than optimal performance for one initial state.

The stable region is identified by a technique borrowed from Pattern recognition, and adapted so as to guarantee that the estimated boundary of the stable region passes through experimentally determined points on that surface. The fact that the technique is simulation based allows it to solve problems which would not be amenable to analytical solutions because of their nonlinearity and complexity. Computer costs do tend to be larger than for analytical techniques, but can be kept at reasonable values. The authors expect to tackle problems of a much larger size than the one given in the example in the near future.

REFERENCES

[1] B. Dubé: "Optimisation des dispositifs de réglage primaire d'une installation hydroélectriques", Bulletin de l'Institut de Réglage Automatique, Ecole Polytechnique Fédérale de Lausanne, No. 76-1 and 76-2, September 1976.

[2] D.E. Kirk: "Optimal Control Theory : An Introduction" (Book), New Jersey, Prentice-Hall, 1970, pp.425-426.

[3] E.J. Davidson and N.S. Rau: "The Optimal Output Feedback Control of a Synchronous Machine", IEEE PAS Vol. 90, No. 4, July/August 1971, pp.2123-2134.

[4] Yao-nan Yu and H.A. M. Moussa: "Optimal Stabilization of a Multi-Machine System", ibid 1975, PAS-94, pp.1174-1182.

[5] T. Athay, R. Podmore, S. Virmani: "A Practical Method for the Direct Analysis of Transient Stability", Paper No. F78 251-1, IEEE PES Winter Meeting, New York, January - February 1978.

[6] D. Crevier: "A Simulation-Based Separation Surface Method for Power System Stability Assessment", Paper No. A78 226-3, IEEE PES Winter Meeting, New York, January - February 1978.

[7] D.F. Specht: "Generation of Polynomial Discriminant Functions for Pattern Recognition", IEEE Trans. on Electronic Computers, June 1967, pp. 308-319.

[8] M.A. Nourmoussavi, D. Crevier: "Multivariable Optimization in a Control System Using a Monte Carlo Technique", Optimization Days, Montreal, May 1978.

[9] M.A. Nourmoussavi: " Power System Control and Stability Using a Separation Surface Technique", Doctoral Dissertation, McGill University, Montreal, Quebec, September 1978.

A MODEL FOR CONSERVATIVE MAGNETIC BUBBLE LOGIC CIRCUITS

Wolfgang Coy

Abteilung Informatik
Universität Dortmund

Dortmund, W.Germany

The first steps in mass production of magnetic bubble memories are done. Now the question of modelling magnetic bubble circuits has become relevant. As magnetic bubble technology is mainly a memory technology and as it is not a replacement for semi-conductor logic, there has been some hesitation against the development of universal sets of basic elements for the construction of bubble logic circuits. In addition, it is apparent that gate-level models of switching circuits are describing inherently continuous signals (such as voltages, currents or even air streams), while magnetic bubbles are discrete entities on a line, not simply a state of a line. Especially the principle of conservation, which reflects the technological difficulty to generate or annihiliate magnetic bubbles in other than especially designed areas of the circuit, has no simple counterpart in gate-level logic.

We show a different model of automata, which allows with a conservative encoding of information the realization of all logical functions (of combinational and/or sequential type) over finite sets of input signals, output signals and states without the necessity to generate or annihiliate bubbles inside the circuit. These circuits may be realized by parallel and feedback connection of only four types of simple automata. These are called convergence, cross-over, fan-out and switch. A switch has two states, two input signals and two output signals; the other basic automata are even simpler. Technical realizations of these basic automata as bubble logic circuits in conservative technique are shown.

Besides this universal set of automata, other universal sets of comparable simplicity are investigated. The 256 candidates for automata with two states, two input and two output signals contain 14 candidates for the replacement of the element "switch". six of them are known to form a universal set together with cross-over, convergence and fan-out (1). For five of them a technical realization as bubble logic circuits is known (2).

(1) Ottmann, Th., Über Möglichkeiten zur Simulation endlicher Automaten durch eine Art sequentieller Netzwerke aus einfachen Bausteinen, Zeitschrift für mathematische Logik und Grundlagen der Mathematik, Vol. 19, pp. 223-238, 1973.

(2) Coy, W., Models for magnetic bubble circuits, (in prep.)

SOME PROPERTIES OF THE NBS DATA ENCRYPTION STANDARD

JOHN P. ROBINSON
Information Engineering
University of Iowa
Iowa City, Iowa 52242

SUMMARY

The DES algorithm is a type of product cyper with an internal product transformation proceeded by an initial permutation and followed by an inverse permutation. The internal transformation consists of 16 key-dependent cycles of bit selection, Module-2 addition and 8 non-linear substitution functions or selection functions. These S-functions are 6 bit to 4 bit functions and are the only non-linear portions of DES.

Various Reed-Muller or ring sum expansions for the S-functions are described and their relationship to the overall operations of DES is explored. The use of an expansion for cryptanalysis is considered.

MINTERM EXCHANGE IN SPECTRAL LOGIC DESIGN

CLAUDIO MORAGA
University of Dortmund, Department of Informatics,
4600 Dortmund 50, Federal Republic of Germany.

ABSTRACT

Minterm exchange is studied in terms of spectral logic design using the Walsh functions. It is shown that minterm exchange preserves certain subsets of spectral elements of the original function. These subsets are simple to identify as zero-vertices of a linear binary function on the same arguments of the logical function. A table-look-up method is suggested to apply these properties to the realization of a large class of logical functions with reduced complexity.

INTRODUCTION

Applications of Harmonic Analysis [1] or Spectral Analysis [2-4] to obtain simple realization of logical functions are already known.

This paper introduces Minterm Exchange (MTE) as an additional operation in spectral logic design using Walsh functions [5] and it is shown that it preserves certain subsets of the original Spectra, from where a table-look-up method for the realization of a large class of logical functions based on a set of simple core functions is deduced. The obtained realizations use at least one exclusive-OR gate, for which low complexity implementations have been reported [6].

NOTATION, DEFINITIONS AND LEMMATA

D1: Let $\underline{Z} = Z_{n-1}Z_{n-2}\ldots Z_1Z_0 \in (0,1)^n$ be a binary n-tuple.

Let $z = \sum_{i=0}^{n-1} Z_i \cdot 2^i$ be a *coding* of $\underline{Z}$. Conversely, let $\underline{Z}$ be the *binary expansion* of z.

Let $f\colon (0,1)^n \to (0,1)$ be a binary function, written $f(z)$.

D2: Let $N = 2^n-1$; let $0 \le w \le N$, with binary expansion $\underline{W}$. The Walsh functions are (here) defined on $[0, 2^n)$, by:

$$X(w,z) = (-1)^{e(w,z)} \quad \text{where} \quad e(w,z) = \sum_{s=0}^{n-1} W_{n-1-s} \cdot Z_s$$

D3: Let $h(v) = (-1)^v$ for all $v \in (0,1)$. It follows that:

$h\colon (\{0,1\}, \oplus) \to (\{1,-1\}, \cdot)$ is an isomorphism, where $\oplus$ stands for modulo 2 addition.

D4: A Transform Pair Function-Spectrum is defined by:

$$S(w) = 2^{-n} \sum_{z=0}^{N} hf(z) \cdot X(w,z)$$
$$f(z) = h^{-1} \sum_{w=0}^{N} S(w) \cdot X(w,z) \tag{1}$$

It has been shown that this pair is unique [4] and will be written as $f(z) \leftrightarrow S(w)$.

D5: Let $f_1(z) \leftrightarrow S_1(w)$ and $f_2(z) \leftrightarrow S_2(w)$ for a given n. Every w such that $S_1(w) = S_2(w)$ will be said to be a *Consistency* of both Spectra.

D6: Let $0 \le z_1, z_2 \le N$. Then

$$P(z_1, z_2) = \{w \mid X(w,z_1) = X(w,z_2)\}$$

is a *Persistence Class*. Similarly, for $0 \le w_1, w_2 \le N$, define

$$P(w_1, w_2) = \{z \mid X(w_1,z) = X(w_2,z)\}$$

D7: Let the following characterizing functions be defined:

$k(w, z_i \oplus z_j) :=$ *if* $(w \in P(z_i,z_j))$ *then* 0 *else* 1

$k(w_r \oplus w_s, z) :=$ *if* $(z \in P(w_r,w_s))$ *then* 0 *else* 1

where $z_i \oplus z_j$ represents the coding of the bitwise modulo 2 addition of the binary expansions of z_i and z_j. Similarly for $w_r \oplus w_s$.

Lemma 1: $(\{k(w,z)\}, \oplus)$ is a Group.

Lemma 2: $(\{k(w,z)\}, \oplus)$ is isomorphic to $(\{X(w,z)\}, \cdot)$ under h.

Proof: To simplify the notation, let $z_i \oplus z_j := z_{ij}$

From Defs. 2, 3, 6 and 7 it follows that:

$$hk(w,z_{ij}) = X(w,z_i) \cdot X(w,z_j) \tag{2}$$

It has been shown [2] that $(\{X(w,z)\}, \cdot)$ is a group such that

$$X(w,z_i) \cdot X(w,z_j) = X(w,z_{ij})$$

which in Eq. (2) leads to

$$hk(w,z_{ij}) = X(w,z_{ij})$$

Now, after Def. 3,

$$\begin{aligned} h(\, k(w,z_{ij}) \oplus k(w,z_{rs})\,) &= hk(w,z_{ij}) \cdot hk(w,z_{rs}) \\ &= X(w,z_{ij}) \cdot X(w,z_{rs}) = X(w,z_{ijrs}) \\ &= hk(w,z_{ijrs}) \end{aligned}$$

from where it follows that

$$k(w,z_{ij}) \oplus k(w,z_{rs}) = k(w,z_{ijrs}) \tag{3}$$

From Eq. (3) it may be concluded that $\oplus$ is a closed operation in the structure $(\{k(w,z)\}, \oplus)$ and it is associative in the structure since it is associative at the index level. Moreover from Eqs. (2) and (3) follows that for all $0 \leq i \leq N$, $k(w,z_{ii}) = k(w,0) = 0$, *i.e.* every $k(w,z)$ is its own inverse. Thus $(\{k(w,z)\}, \oplus)$ is a group whose identity is $k(w,0)$. From Eq. (2) it may be seen that h is a one to one mapping from $(\{k(w,z)\}, \oplus)$ onto $(\{X(w,z)\}, \bullet)$, thus, both groups are isomorphic.

Corollary 3: $(\{k(w,z)\}, \oplus)$ is isomorphic to $(\{L(w)\}, \oplus)$, where $\{L(w)\}$ is the set of linear functions on w.

Proof: It has been shown [2] that h is an isomorphism from $(\{L(z)\}, \oplus)$ onto $(\{X(w,z)\}, \bullet)$. Since from Def. 2 $X(w,z) = X(z,w)$ then $(\{L(z)\}, \oplus)$ is isomorphic to $(\{X(z,w)\}, \bullet)$ and $(\{L(w)\}, \oplus)$ is isomorphic to $(\{X(w,z)\}, \bullet)$. The corollary follows directly from Lemma 2.

Corollary 4: There **are** 2^n different Persistence Classes $P(z_i,z_j)$. Every Persistence Class $P(z_i,z_j)$ with $i \neq j$, has 2^{n-1} elements, where n is the number of arguments of the corresponding logical function.

Proof: follows directly from corollary 3.

Lemma 5: Let $f(z)$ be a boolean binary function. Let $f'(z)$ be the function obtained from $f(z)$ after exchanging two (different) minterms. Let $f(z) \leftrightarrow S(w)$ and $f'(z) \leftrightarrow S'(w)$. Then:

i) $S(w)$ and $S'(w)$ have 2^{n-1} Consistencies

ii) There exists a $L(w)$ such that for all above mentioned Consistencies $L(w) = 0$.

Proof: Assume that z_i and z_j represent the minterms to be exchanged.

Let

$$R(w) = p^{-n} \sum_{z=0}^{N} hf(z)\bullet X(w,z)\bullet \delta(z)$$

where $\delta(z) :=$ *if* $(z=z_i$ *or* $z=z_j)$ *then* 0 *else* 1.

Then:

$$S(w) = R(w) + p^{-n}hf(z_i)\bullet X(w,z_i) + p^{-n}hf(z_j)\bullet X(w,z_j)$$

and $$S'(w) = R(w) + p^{-n}hf(z_j)\bullet X(w,z_i) + p^{-n}hf(z_i)\bullet X(w,z_j)$$

From Def. 6 and Corollary 4, $P(z_i,z_j)$ has 2^{n-1} elements w such that $X(w,z_i) = X(w,z_j)$, which proves assertion (i).

Assertion (ii) follows directly from Corollary 3.

Lemma 6: Exchange of one or more minterm-pairs belonging to a same Persistence Class leads to a same set of Consistencies.

Proof: follows directly from Def. 6.

APPLICATIONS TO SPECTRAL LOGIC DESIGN

One of the goals of spectral logic design is to achieve low complexity realizations of logical functions. Usually the resulting design consists of a "simple" core and a peripherial circuitry which realizes some appropriate transformation on the core function. The simplest transformations are: complementation of some arguments, permutation of arguments and complementation of the function, which have the following effect upon the spectrum of the logical function [1, 2]:

If $f(z) \leftrightarrow S(w)$

then: $\bar{f}(z) \leftrightarrow -S(w)$ (4)

$f(z \oplus a) \leftrightarrow X(w,a) \cdot S(w) \quad , \quad 0 \le a \le N$ (5)

$f(A \otimes z) \leftrightarrow S(w \otimes A')$ (6)

where A is a non singular orthogonal binary matrix which induces argument permutation, A' is obtained from A^{-1} after inverse reordering of both lines and columns and $(A \otimes z)$ means the coding after the modulo 2 matrix product between A and the transposed binary expansion of z considered as a vector. Similarly for $(w \otimes A')$. Additional transformations have been studied in references [1 - 4].

Methods for the "optimized" logic design range from functional analysis on some abstract complexity criteria [2] to simple spectral inspection and table-look-up procedures [3] .

A table based method of applying MTE to logic design will be discussed below.

Assume that a set of simple functions has been selected to constitute the core. Assume that the corresponding Spectra have been calculated and listed together with the core functions. Then the following algorithm may be applied for the MTE realization test of a given function f(z):

(i) From f(z) calculate S(w). (Fast algorithms are known [7]).

(ii) Check the table of core functions and corresponding Spectra looking for a (transformable) S'(w) which has 2^{n-1} consistencies with S(w), such that the cluster of consistencies cover the zeroes of some L(w). If no S'(w) is found, the given function has no realization by MTE on the selected core. If however an S'(w) is found, then f'(z) may be read (calculated) from the table.

(iii) Calculate $f''(z) = f(z) \oplus f'(z)$, from where $f(z) = f'(z) + f''(z)$ is the suggested realization. Notice that f"(z) induces MTE by local complementation. (An analysis of MTE as a permutation upon f(z) and its *modifying* effect upon the spectrum of f(z) has been earlier reported [8], but no applications to logic design are known from that approach).

(iv) Test for complexity. (Even though the test for complexity does not properly belong to the MTE realization algorithm, it has been included on purpose, to emphasize that no claim is being stated that MTE should *always* produce an "optimal" realization.)

EXAMPLE

Let f(z) be a 4-place binary function as shown in Fig. 1a. (Notice that instead of Karnaugh maps with Gray coding, maps with Dual coding are used [9]. Similar maps will also be used to represent the Spectrum of the function.). By some classical method it is simple to show that f(z) has the following boolean expression:

$$f(z) = Z_1 \vee Z_3\bar{Z}_2 \vee Z_3\bar{Z}_0 \vee \bar{Z}_3Z_2\bar{Z}_0 \qquad (7)$$

which suggests the realization shown in Fig. 2a.

By simple inspection, it is possible to give a more compact expression to f(z), namely

$$f(z) = Z_1 \vee (Z_3 \oplus Z_2\bar{Z}_0) \qquad (8)$$

which leads to the realization shown in Fig. 2b.

What type of realization may be obtained by MTE ?

Assume the required catalog of selected core functions is available.

Following the formerly described algorithm:

(i) From f(z) calculate S(w). The map of 16 S(w) is shown in Fig. 1a.

(ii) Check the catalog. An S'(w) as shown in Fig. 1b should be found, which has 8 Consistencies with S(w) occuring on $L(w) = W_0 = 0$. The corresponding f'(z), shown in Fig. 1b may directly be recognized to be $Z_1 \vee Z_3$. (Such a simple function would certainly be a member of any reasonable catalog of core functions.)

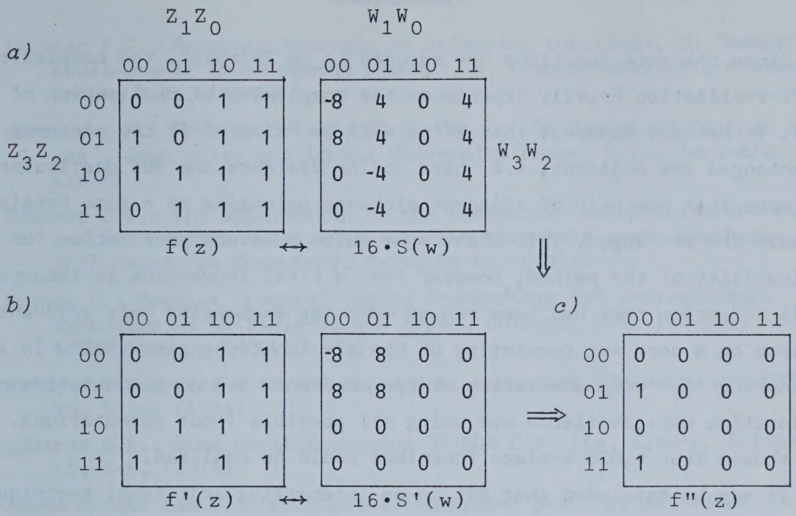
(iii) f"(z) may be obtained by elementwise modulo 2 addition of f(z) and f'(z), giving the function shown in Fig. 1c, from where it may be read that $f''(z) = Z_2\bar{Z}_1\bar{Z}_0$.

It follows that

$$f(z) = f'(z) \oplus f''(z) = (Z_1 \vee Z_3) \oplus Z_2\bar{Z}_1\bar{Z}_0 \qquad (9)$$

leading to the realization shown in Fig. 2c.

(iv) By simple inspection of Fig. 2 it may be seen that in this case the MTE-realization is much *simpler* than the one suggested in Eq. (7) and *faster* than the one suggested in Eq. (8).



a)

Z_3Z_2 \ Z_1Z_0	00	01	10	11
00	0	0	1	1
01	1	0	1	1
10	1	1	1	1
11	0	1	1	1

f(z) ↔

W_3W_2 \ W_1W_0	00	01	10	11
00	-8	4	0	4
01	8	4	0	4
10	0	-4	0	4
11	0	-4	0	4

16•S(w)

⇓

b)

	00	01	10	11
00	0	0	1	1
01	0	0	1	1
10	1	1	1	1
11	1	1	1	1

f'(z) ↔

	00	01	10	11
00	-8	8	0	0
01	8	8	0	0
10	0	0	0	0
11	0	0	0	0

16•S'(w) ⟹

c)

	00	01	10	11
00	0	0	0	0
01	1	0	0	0
10	0	0	0	0
11	1	0	0	0

f"(z)

Fig. 1: Map representation of functions and spectra involved in the design example.

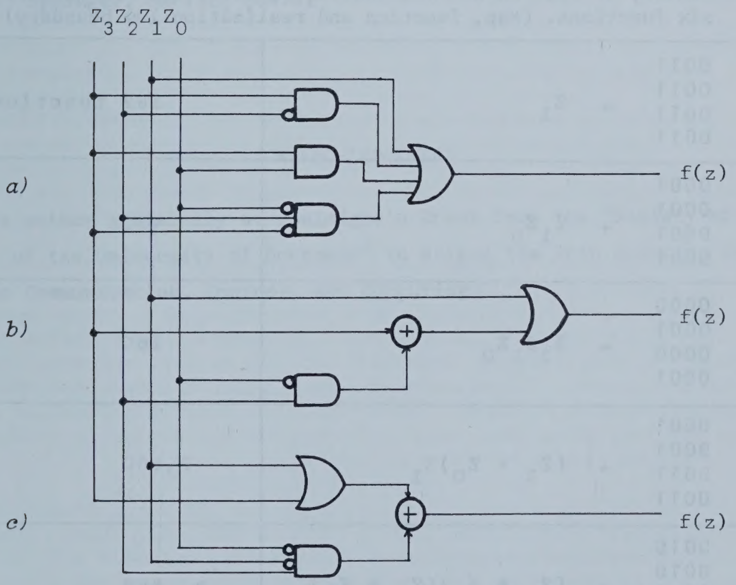


Fig. 2 *a*, *b* and *c*: Logic diagram of realizations after Eqs. (7), (8) and (9) respectively.

DISCUSSION

Since the core functions are assumed to be "simple", the complexity of an MTE-realization heavily depends on the complexity of realization of f"(z). It becomes apparent that f"(z) will be "simple" if the minterms to be exchanged are adjacent, *i.e.* have coding distance one and particularly, when more than one pair of adjacent minterms belonging to a same Persistence Class are exchanged. This might seem quite a severe restriction for the applicability of the method, however such a first impression is incorrect. A full induction test has been run by computer exchanging only adjacent minterms on a core set consisting of the six functions given below in Table 1, assuming that complementation of the arguments and complementation of the function were available and using all possible input permutations. The test showed that 4,030 4-place functions could be realized.

It may be concluded that MTE is an interesting additional technique to be used in spectral logic design, providing an *alternative* which seems worthwhile to be considered and further developed.

Table 1: Summary of Test on MTE realization capabilities over a core of six functions. (Map, function and realization low boundary)

0011 0011 0011 0011 → Z_1	182 functions
0001 0001 0001 0001 → Z_1Z_0	432
0000 0001 0000 0001 → $Z_2Z_1Z_0$	160
0001 0001 0011 0011 → $(Z_3 \vee Z_0)Z_1$	2,560
0010 0010 0100 0100 → $(Z_3 \oplus Z_0)(Z_1 \oplus Z_0)$	≥ 448
0110 0110 1001 1001 → $Z_3 \oplus Z_1 \oplus Z_0$	≥ 248

REFERENCES

1. Lechner R.J.: Harmonic Analysis of switching functions, in *"Recent Developments in Switching Theory"*, A. Mukhopadhyay Ed., Academic Press, N.Y., (1971)
2. Karpovsky M.: *"Finite Orthogonal Series in the Design of Digital Devices"*, John Wiley and Israel University Press, N.Y., Jerusalem, (1976)
3. Edwards C.R.: The application of the Rademacher-Walsh Transform to Boolean function classification and threshold logic synthesis, *IEEE Trans. on Computers*, C-24, 48-62, (1975)
4. Moraga C.: Complex Spectral Logic, *Proceedings 8th International Symposium on Multiple-valued Logic*, Chicago, USA, 149-156, (1978) (IEEE Catalog 78CH1366-4C)
5. Walsh J.L.: A closed set of orthogonal functions, *American Jr. Math.*, 45, 5-24, (1923)
6. Edwards C.R.: Some Novel Exclusive OR/NOR Circuits, *Electr. Letters*, 11, 3-4, (1975)
7. Ahmed N. and Rao K.: *"Orthogonal Transforms for Digital Signal Processing"*, Springer, Berlin, (1975)
8. Eris E.: A minterm interchange operation in the Walsh Domain, *Electr. Letters*, 14, 92-94, (1978)
9. Giloi W. and Liebig H.: *"Logischer Entwurf digitaler Systeme"*, Springer, Berlin, (1973)

ACKNOWLEDGEMENT

The author gratefully acknowledges a Grant from the "Society of Friends of the University of Dortmund" to attend the 16th Allerton Conference on Communication, Control, and Computing.

A UNIQUE INTERFACE TO IMS

JOHN J. MAYEWSKI
IBM Corporation
Sterling Forest, New York

ABSTRACT

The Common Manufacturing Information System (CMIS) is a conglomerate of seven software applications dedicated to controlling the manufacturing processes of three domestic and two IBM European locations. There are six applications written in Basic Assembler Language and one written in PL1. They are Procurement (PROC), General Information Retrieval (GIR), Manufacturing Activity Release and Control (MARC), Final Assembly Logistics Control (FALC), Warehousing (WHSE) and Procurement Quality Assurance (PQA). The application written in PL1 is the Development Production Records System (DPRS) which currently runs with Information Management System under Virtual Systems control and uses an IMS data base.

The first six applications run under control of a data base, data communications system known as DACS/TSPS. DACS, or Data Access Control System, runs under control of Operating System or Virtual System 360/370 and is the data base interface for all applications. TSPS is Telecommunications Support Processor System, an operating system in itself. It is the terminal network interface to DACS. These combined systems perform similar functions to that of an IMS data base/data communications system.

An interface had to be developed between TSPS and IMS so that DPRS users could access the DACS data base of the Product and Process Description (PPD) application. DPRS is the rewritten version of PPD. Running both versions allowed a gradual migration and education by location users.

The way in which the interface was developed is what makes it so unique. Normally, the TSPS system is the slave of DACS. When processing is required, the data base system will solicit TSPS for transactions to be processed. That philosophy was carried through in developing the TSPS-IMS Interface.

The TSPS system appears to be a 2770 terminal system to IMS. The reason for this is because both systems (TSPS and the 2770) can support a variety of terminal types. The TSPS development group designed a device dependent module (DDM) which replaced the 2770 DDM link edited in the IMS nucleus. This module provides the same interface into IMS as does any other DDM with a few exceptions. These apply to manipulating the data into IMS format or blocking it for TSPS. There are other modules and tables with which the DDM interfaces that were developed for the project and will be discussed in more detail in the completed paper.

One of the IMS interface requirements made by the IBM World Trade Corporation was that there would be no modification to IMS code. This restriction made the Interface design even more difficult, but also allowed each site, that was to use this interface, to receive continued support from IMS Program Support Representatives. That restriction, however, did not totally prevent the Interface from becoming IMS level sensitive. In fact, each release, including Program Temporary Fixes (PTF's) must be tested to expose any possible problems and ensure data integrity.

A FAST REARRANGEMENT TECHNIQUE ON THE ROW-COLUMN INDEXING SCHEME FOR THE SOLUTION OF SPARSE SYSTEMS OF EQUATIONS

EIHACHIRO NAKAMAE*, HIDEO YAMASHITA*, AND TOSHIHIDE NOMURA°

*Department of Electrical Engineering
Faculty of Engineering
University of Hiroshima
Hiroshima 730, Japan

°Department of Electrical
Engineering
Kure Technical College
Kure 737, Japan

ABSTRACT

A new method of rearranging the data of an *n*×*n* matrix on the row-column indexing scheme is presented, where row or column numbers and their contents are separated into *n* groups and they are dispersed in the one dimensional array. The results of some examples applied to the finite element analysis show that the processing time for rearrangement becomes greatly shortened; the time ratio of the proposed method and traditional method is only 1 to 35 in the case of *n* is 445 for example.

I. INTRODUCTION

Processing programs solving sparse systems of equations should be evaluated from the point of view of computational and storage savings. Recently several papers$^{1)\sim3)}$ on optimal pivot ordering algorithms have been published. It is also a very important subject to find a suitable data structure and its skillful processing method from the viewpoint mentioned above.

A row-column indexing scheme is very often used in the processing programs to solve sparse systems of equations because of its usefulness$^{4)}$; in this scheme the row or column numbers of nonzero elements in a matrix and their contents are stored in one dimensional arrays, respectively, and the rearrangement of their order is required on each step of pivot ordering because of fill-in. Since the operation for the rearrangement takes much time, shortening this process is one of the important subjects.

In this paper a new rearrangement method on the row-column indexing scheme is proposed; row or column numbers and their contents are separated into *n* groups, where *n* shows the dimension of the matrix and the groups are dispersed in the one dimensional arrays, even though in the traditional row-column indexing scheme they are concentrated in the front part of the arrays. The method proposed in this paper, however, needs the extra locations of which number is equal to *n*. This method is applied to a processing program to solve symmetric simultaneous linear equations using Choleski's method. The results show that the processing time for rearrangement becomes very short; for example, in the finite elements analysis the time reduces to 1/25 and 1/35 in the cases of *n* is 229 and 445, respectively, as compared with the traditional method.

II. ASSIGNATION OF BLANK SPACES

Figure 1 shows one of examples of data structures of traditional row-column indexing schemes. In this case the operation is executed in row manner: three arrays consist of a row pointer array, IR, a column array, IC, and an array storing the contents of nonzero elements, VALUE, where IC and VALUE do not include the diagonal elements, and the contents of the diagonal elements in nonzero elements are stored in the other one dimensional array.

When a fill-in is created on the step of pivot ordering, its information should be entered into IC and VALUE and every nonzero element behind

	1	2	3	4	5
1	a_{11}		a_{13}		a_{15}
2		a_{22}		a_{24}	a_{25}
3	a_{31}		a_{33}	a_{34}	Δ
4		a_{42}	a_{43}	a_{44}	a_{45}
5	a_{51}	a_{52}	Δ	a_{54}	a_{55}

	1	2	3	4	5	6
IR	1	3	5	7	10	13

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
IC	3	5	4	5	1	4	2	3	5	1	2	4										

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
VALUE	a_{13}	a_{15}	a_{24}	a_{25}	a_{31}	a_{34}	a_{42}	a_{43}	a_{45}	a_{51}	a_{52}	a_{54}										

Figure 1. A traditional row-column indexing scheme

the locations of these entries should be moved backward one by one. For example, in Figure 1 if a_{11} is chosen as a pivot, the fill-ins are created at the locations, (3,5) and (5,3), as shown by the sign Δ. Then, the operations shown in Figure 2 are executed. In general, the greater the increase of the dimensions of a matrix, the greater the number of rearrangement operations. When the operation of pivot ordering has been finished, the memory capacity of IC and VALUE needs at least the number of nonzero elements, L_0, where the diagonal elements are excepted. The value of L_0, however, is unknown before pivot ordering, therefore the number of locations, L, whose value is larger than L_0, is set for IC and VALUE. The method proposed in this paper makes use of the blank spaces whose number is $L-N_f$, where N_f is given by equation (1): the total number of nonzero elements except the diagonal elements before pivot ordering; that is, before pivot ordering the information in IC and VALUE is separated into groups consisting of each row's elements, and the groups are dispersed in the same arrays. For example, in the case of Figure 1 if two blank spaces are set on each row, the data structure is expressed by Figure 3. The number of operations for rearrangement is reduced as shown in Figure 4, but an extra pointer array, IRE, which points out the end points of each row in IC and VALUE, is required.

1) IC(14)=IC(12) VALUE(14)=VALUE(12)
2) IC(13)=5 VALUE(13)=a_{35}
3) IC(12)=IC(11) VALUE(12)=VALUE(11)
4) IC(11)=IC(10) VALUE(11)=VALUE(10)
5) IC(10)=IC(9) VALUE(10)=VALUE(9)
6) IC(9)=IC(8) VALUE(9)=VALUE(8)
7) IC(8)=IC(7) VALUE(8)=VALUE(7)
8) IC(7)=3 VALUE(7)=a_{53}

Figure 2. Operation procedures to insert fill-in created

The blank spaces should be assigned in order to save the number of operations for rearrangement. The purpose of pivot ordering algorithms used for the solution of sparse systems of equations is to minimize the creation of fill-ins. In these algorithms the rows having many nonzero elements before pivot ordering have a high probability of creating fill-ins. The blank spaces are, therefore, assigned in proportion to the number of nonzero elements in each row before pivot ordering.

If in an $n \times n$ matrix the number of nonzero elements in the ith row before the pivot ordering is n_i, N_f is expressed by the following equations;

	1	2	3	4	5
IRS	1	5	9	13	18
IRE	2	6	10	15	20

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
IC	3	5			4	5			1	4			2	3	5			1	2	4		

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
VALUE	a_{13}	a_{15}			a_{24}	a_{25}			a_{31}	a_{34}			a_{42}	a_{43}	a_{45}			a_{51}	a_{52}	a_{54}		

Figure 3. Assignation method of blank spaces

$$N_f = \sum_{i=1}^{n} n_i - n, \qquad (1)$$

then the number of blank spaces in the ith row, m_i, is given by the following equation;

1) IC(11)=5 VALUE(11)=a_{35}
2) IC(21)=IC(20) VALUE(21)=VALUE(20)
3) IC(20)=3 VALUE(20)=a_{53}

Figure 4. Operation procedures to insert fill-in created

$$m_i = \left[\frac{L - N_f}{N_f} \cdot (n_i - 1) \right], \qquad (2)$$

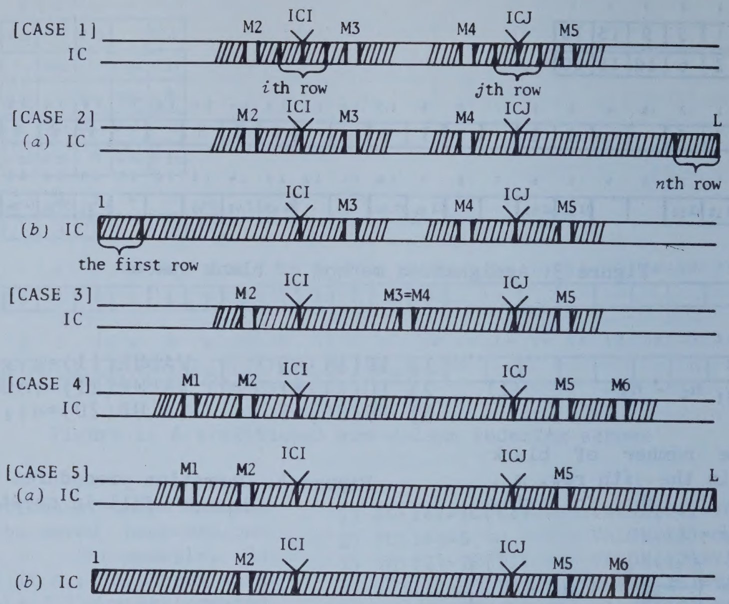
where the sign [ ] is Gaussian.
The halves of m_i blank spaces are assigned on both sides of nonzero elements in the ith row, respectively, because the operations of rearrangement are executed in both directions in order to minimize the number of operations to remove the contents of nonzero elements.

III. REARRANGEMENT ALGORITHM

An algorithm to minimize the number of operations for rearrangement on each step of pivot ordering is proposed, where it is assumed that the matrix discussed in this paper is symmetric and the column pointer array IC has the blank spaces proposed in Chapter II. Since the arrays IC and VALUE are homologous, we discuss the algorithm of the rearrangement of IC only.

When a fill-in is created, two nonzero elements are inevitably created on the symmetrical locations because of the symmetric matrix. For example, let's assume that if these elements are created at (i,j) and (j,i), where $j>i$, this information is entered into ICI and ICJ in IC, respectively, where ICI<ICJ. According to the relationship of locations between ICI and ICJ and blank spaces, the blank spaces minimizing the number of operations for arrangement should be chosen as the candidates. All cases of the relationships of the locations are six; five cases are shown in Figure 5, and the other is one in which only two blank spaces exist in IC. In the case of the latter there is no choice, and if the blank space is only one or zero, the memory of IC overflows.

In Figure 5, M2 and M3 are the nearest blank spaces from ICI in decreasing and increasing directions, respectively; M4 and M5 are the nearest blank spaces from ICJ in decreasing and increasing directions, respectively; M1 and M6 are the second nearest blank spaces from ICI in a decreasing direction and from ICJ in an increasing direction, respectively. All candidates of each pair in Figure 5 are expressed by Table 1; a pair


[CASE 1]
IC
M2
ICI
M3
M4
ICJ
M5
*i*th row
*j*th row
[CASE 2]
(*a*) IC
M2
ICI
M3
M4
ICJ
L
*n*th row
1
(*b*) IC
ICI
M3
M4
ICJ
M5
the first row
[CASE 3]
IC
M2
ICI
M3=M4
ICJ
M5
[CASE 4]
IC
M1
M2
ICI
ICJ
M5
M6
[CASE 5]
(*a*) IC
M1
M2
ICI
ICJ
M5
L
1
(*b*) IC
M2
ICI
ICJ
M5
M6

(The shaded areas show nonzero elements.)

Figure 5. Relative relationships of the locations between ICI and ICJ and blank spaces

chosen from them should have the smallest number of operations for rearrangement in each case. The pairs in Table 1 consist of couples shown in Table 2 thus the number of operations for rearrangement in IC can be counted by using Table 2. The algorithm to choose the pair of blank spaces is as follows; where the locations of six blank spaces M_i (i = 1∿6) in Figure 5 are stored in MAG(i) (i = 1∿6).

[step 1] Set MAG(I) ⟵ -2×L (I = 1∿6), MX ⟵ 2, and NM ⟵ 0 ; go to the next step.

[step 2] Search a blank space from the *i*th row toward the *j*th row; if there is a blank space, set MAG(3) ⟵ its location in IC, MX ⟵ 1, and NM ⟵ NM+1, and go to the next step; if there is no blank space, go to step 5.

Table 1. Candidates of the pairs of the blank spaces for rearrangement

CASE 1		(M2, M4), (M2, M5), (M3, M4), (M3, M5)
CASE 2	(*a*)	(M2, M4), (M3, M4)
	(*b*)	(M3, M4), (M3, M5)
CASE 3		(M2, M4(= M3)), (M2, M5), (M3(= M4), M5)
CASE 4		(M1, M2), (M5, M6), (M2, M5)
CASE 5	(*a*)	(M1, M2), (M2, M5)
	(*b*)	(M2, M5), (M5, M6)

Table 2. Number of operations for rearrangement

k	(M_i, M_j)	IDO(k)
1	(M1, M2)	IDO(1) = ICJ - M1 - 2
2	(M2, M4)	IDO(2) = |ICI - M2 - 1| + |ICJ - M4 - 1|
3	(M2, M5)	IDO(3) = |ICI - M2 - 1| + |ICJ - M5|
4	(M3, M4)	IDO(4) = |ICI - M3| + |ICJ - M4 - 1|
5	(M3, M5)	IDO(5) = |ICI - M3| + |ICJ - M5|
6	(M5, M6)	IDO(6) = |ICI - M6 + 1|

[step 3] Search a blank space from the jth row toward the ith row, set MAG(4) ← its location in IC, and go to the next step.

[step 4] If MAG(3) is equal to MAG(4), go to the next step; if MAG(3) is not equal to MAG(4), set NM ← NM+1 and go to the next step.

[step 5] Search MX blank spaces from the ith row toward the first row; if there is no blank space, go to the next step; if the blank spaces are MX', where $MX \geq MX' > 0$, set MAG(3-k) ← the locations of them in IC, where $k = 1 \sim MX'$, and NM ← NM+MX', and go to the next step.

[step 6] Search MX blank spaces from the jth row toward the nth row; if there is no blank space, go to the next step; if the blank spaces are MX', where $MX \geq MX' > 0$, set MAG(4+k) ← their locations in IC, where $k = 1 \sim MX'$, and NM ← NM+MX', and go to the next step.

[step 7] If $NM \leq 1$, stop the calculation and type the error because of shortage of memory capacity; if $NM \geq 2$, go to the next step.

[step 8] Calculate IDO(k) by using Table 2, where $k = 1 \sim 6$; search a pair (M_i, M_j) of which IDO(k) is one of the minima; stop.

IV. EXAMPLES AND DISCUSSION

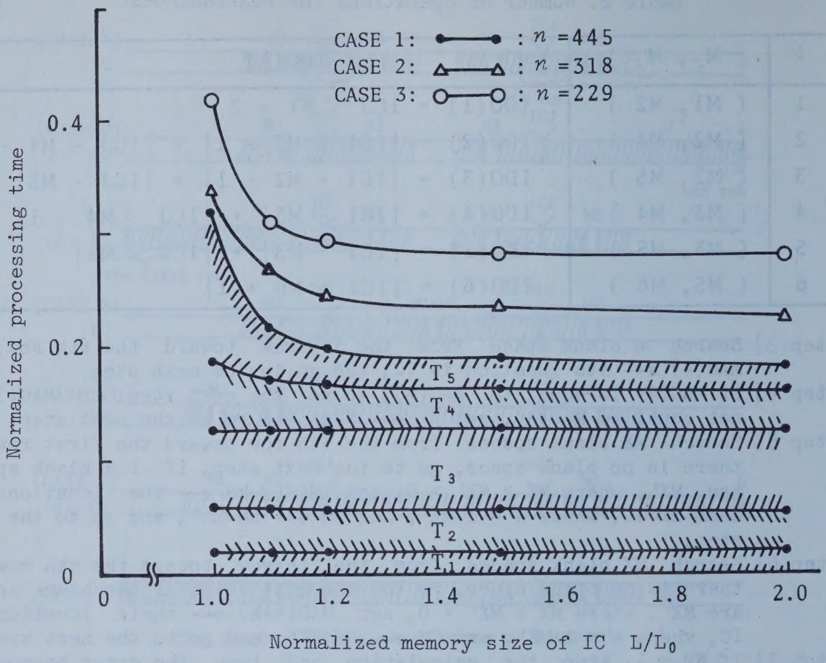
The method proposed in this paper is compared with a traditional method in which nonzero elements are concentrated in the front part of a one dimensional array.

For comparisons between them a processing program to solve symmetric simulteaneous linear equations using Choleski's method is applied; the program is used to obtain flux distributions by using the finite element method, where in the pivot ordering algorithm the candidates of pivots are only diagonal elements. The pivots are chosen by Markowitz algorithm,[1) and if there are several candidates, an element whose absolute value is the maximum is chosen.

1] Comparisons between Proposed Method and Traditional Method

In order to compare the two methods, it is assumed that the total processing time to solve a equation and the memory capacity L_0 in the traditional method are set as a unit. Figure 6 demonstrates the relationship between memory sizes of IC and processing time, and the items of processing time in case 1. The several results of the traditional method are shown in Table 3 in detail. The computer used is HITAC 8700.

As a natural result the time T_1, T_2, and T_3 are not affected by memory sizes and differences of the two methods. The figure shows that in case 1 when the traditional method is used the amount time of T_1, T_2, and T_3 is only 13 percent of the whole processing time; it means that most of the



T_1 is the time for the decision of pivot ordering.
T_2 is the calculation time of the forward elimination and the backward substitution.
T_3 is the time of examination if fill-in is or is not created and the time for the decision of the position if fill-in is created.
T_4 is the time for searching for the suitable pairs of blank spaces.
T_5 is the time for the rearrangement.

Figure 6. Relationship between memory sizes of IC and processing time

Table 3. Results by traditional method

n	N_f	L_0	F_n	T_t (Sec)	IDO (Times)
445	1749	7385	5636	67.0	9.52×10^6
318	1306	4596	3290	27.3	3.70×10^6
229	1067	3621	2554	16.9	2.20×10^6

F_n : Number of fill-ins created T_t : Total processing time
IDO : Number of rearrangement operations in IC

processing time, 87 percent, is taken for the operation time for rearrangement. While using the method proposed in this paper the rearrangement processing time consists of T_4 and T_5; T_4 is for searching for the suitable pair of blank spaces, and T_5 is for movement of nonzero element within the array; in the traditional method T_4 is zero. The figure shows that in case 1 when the proposed method is used, even if $L = L_0$, $T_4 + T_5 = 0.2$, and if $L/L_0 = 1.2$, $T_4 + T_5 = 0.07$; the result of the latter means that the rearrangement processing time is shortened to less than one twelfth as compared with the traditional method. If $L = L_0$, the method proposed takes about one third of the whole processing time of the traditional method; if $L/L_0 = 1.2$, it takes one fifth.

In practice before calculation L_0 is generally unknown, therefore L must be set to at least more than 1.3 times of L_0; it means that the method proposed can drastically shorten the processing time with almost the same memory capacity as that of the traditional method. Figure 6 also shows that the more the dimensions of a matrix expand, the more its efficacy increases.

2] Assignation of Blank Spaces

The following three methods are compared.

《Method 1》 This method was proposed in Chapter II; blank spaces are assigned in proportion to the number of nonzero elements in each row before pivot ordering.

《Method 2》 The number of blank spaces assigned into the ith row, m_i, is given as follows:

$$m_i = \left[\frac{L - N_f}{n} \right] \qquad (3)$$

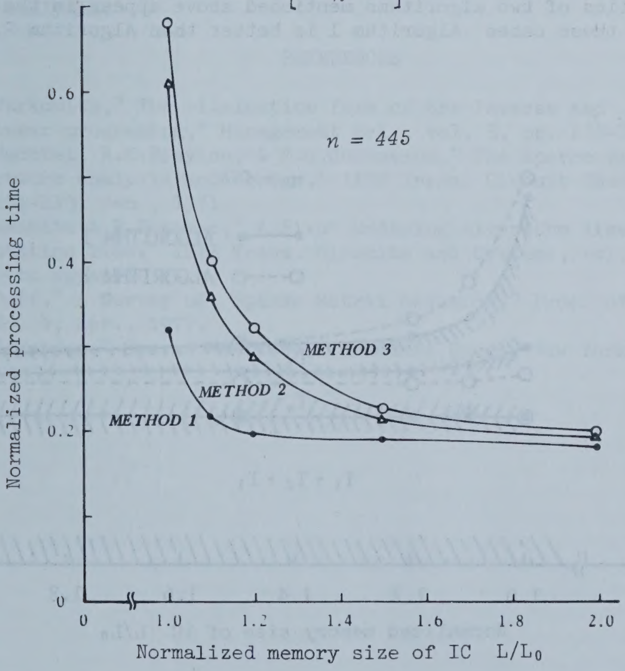


Figure 7. Comparison of assignation methods of blank spaces

((Method 3)) m_i is given as follows:

$$m_i = \left[\frac{L}{n}\right] - n_i \tag{4}$$

In all cases each half of m_i is assigned on both sides of nonzero elements in the ith row, respectively.

The results in Figure 7 show that the Method 1 is excellent compared with the other two methods because of the reason discussed in Chapter II. When the ratio of L to L_0 is near to a unit, it is remarkable.

3] Rearrangement Algorithm

The following two algorithms to rearrange the data of IC are compared.

((Algorithm 1)) This algorithm was proposed in Chapter III. In order to minimize the number of operations for rearrangement, data are removed in two directions, the forward and backward directions.

((Algorithm 2)) Data are removed backward only as far as possible, and only if there is no blank space backwards, the data are removed forward.

A strong point of Algorithm 1 is that the number of operations for rearrangement might be decreased, while a weak point is that the time to search suitable pairs of blank spaces might be large. Algorithm 2 has contrary characteristics. The sizes of both programs are nearly equal. Figure 8 compares the two algorithms. The blank spaces are assigned by using Method 1, and in the case of Algorithm 2, m_i blank spaces obtained by equation (2) are assigned behind the nonzero elements of the ith row. The characteristics of two algorithms mentioned above appear in the figure, and at least in these cases Algorithm 1 is better than Algorithm 2, but it is not serious.

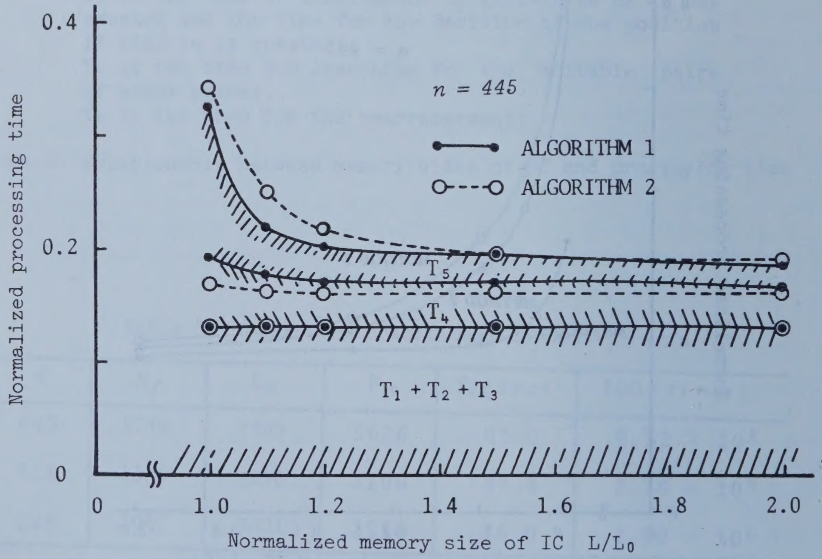


Figure 8. Comparison of rearrangement algorithm

V. CONCLUSION

In this paper a new method of scatteringly assigning blank spaces in both arrays, a column pointer array and an element array, of a row-column indexing scheme, is proposed. It was shown that the method is very useful for computational saving by applying a processing program to solve symmetric simultaneous linear equations using Choleski's method; the examples showed that the larger the dimensions of a matrix, the higher the efficiency. The method, of course, can be applied to general processing programs for nonsymmetric matrices.

Another typical data structure for sparse matrices is a list scheme[5]. A strong point of this is that when fill-ins are created, it is enough to rewrite some pointers; while a weak point of this is that when the access of elements is executed in order row or column direction, the processing time of a list scheme is greater than that of a row-column indexing scheme. If a computer has a virtual memory and the main internal memory capacity is relatively small, in a list scheme the access time for equation solving increases extremely due to the increase in the time for bringing the data into the main internal memory from the secondary memory. Under the conditions mentioned above, a row-column indexing scheme is more suitable than a list scheme as a data structure used for a sparse matrix processing program. In the case of a program written in FORTRAN language, the memory capacity of the latter increases to at least about 1.5 times as compared to the former.

The method proposed in this paper might be applied to programs in which data to keep a particular order is required and nonzero elements are created at random; for example, it might be effective to use the proposed method in order to make up a coefficient matrix of equations in a finite element analysis.

REFERENCES

1) H.M.Markowitz," The elimination form of the inverse and its application to linear programing," Management Sci., vol. 3, pp. 255-269, Apr., 1957.
2) G.D.Hachtel, R.K.Brayton, & F.G.Gustavson," The sparse tableau approach to network analysis and design," IEEE Trans. Circuit Theory, vol. CT-18, pp. 101-113, Jan., 1971.
3) H.Yamashita & E.Nakamae," A Pivot Ordering Algorithm Aimed at Minimizing Computation Time," IEEE Trans. Circuits and Systems, vol. CAS-25, No. 8, 1978.(to appear)
4) I.S.Duff," A Survey of Sparse Matrix Research," Proc. of the IEEE, vol. 65, No. 4, Apr., 1977.
5) R.P.Tewarson," Sparse Matrices," Academic Press, New York, 1973.

A NATIONAL SYSTEMS CENTER PROPOSAL

STEPHEN KAHNE
Case Western Reserve University
University Circle
Cleveland, Ohio 44106

ABSTRACT

This paper contains the outline of a proposal for the development of a national systems center (NSC). This proposal has resulted from deliberations of a committee called, in this paper, the Ad Hoc NSC Committee, chaired by Professor Michael Athans of MIT. Twenty-three members of this committee have been particularly active in developing this proposal. Another twenty-five committee members played more minor roles in the development. This paper will outline the current thinking of the committee and is written in an attempt to solicit suggestions rather than present the final version of the plan.

CONCLUSIONS AND RECOMMENDATIONS

The nation faces a sequence of critical decisions with respect to current and future complex technical, economic, and social issues. Limited resources require the coordination and optimization of complex interconnected, large scale, dynamic, stochastic systems. Existing methodologies are not sufficiently advanced, nor are they holistic enough, to provide understanding to inform policy-makers at either the strategic or tactical levels. A significant national need exists for basic and applied research in complex large scale dynamic and stochastic systems. The available intellectual resources of the country are too fragmented by existing departmental and institutional barriers to address the complex and long range national needs and to provide technical advice to decision makers in matters of policy analysis and synthesis.

It is recommended that a high-quality, independent, non-profit national systems center be created to conduct basic and applied research in the general areas of large scale systems analysis, modeling methodologies, and tools for policy analysis and synthesis. Such a center would constitute a national asset which is urgently needed.

A HISTORICAL PERSPECTIVE OF NSC DISCUSSIONS

The Workshop on New Directions in Systems Science and Engineering was held in Ann Arbor in 1972. The output of this Workshop was a recommendation to the National Science Foundation to consider formation of a National Systems Center. This idea was further discussed in 1975 at a NSF-sponsored Workshop on Modeling Large Scale Systems at National and Regional Levels at the Brookings Institution in Washington, D.C. In 1976, NSF sponsored the National Center for Systems Analysis Workshop at Ann Arbor. At this workshop an ad hoc committee was elected to conduct further studies into the possibility of a National Systems Center. Several successive meetings occurred and the membership of the ad hoc committee was enlarged to include a broader spectrum of people interested in this matter. On May 12, 1978, a workshop was held at MIT and a presentation was made which is essentially identical to the present paper. About 80 people attended this workshop and still additional input was obtained concerning the viability and feasability of a national systems center.

THE SYSTEMS NATURE OF NATIONAL PROBLEMS

The systems problems at the national level may be divided into physical systems, sociotechnical systems, and information systems. Energy, food, material and environmental issues present problems of finite resources and environmental capacity that must be dealt with while taking into account economic and political issues. As constraints become more binding, local subsystems become more interconnected. Local optimization can lead to very inefficient operation and even instability if the degree of interconnection and interdependence is increased. Interconnected large scale systems can exhibit poorly understood behavior of a dynamic and stochastic nature. Although systems science and engineering has begun to make contributions to some of these areas the systems science discipline has not matured to a level where it can be used routinely to contribute to the solution of these complex problems. In addition, there is a need to increase public awareness so that these large scale problems are viewed in their appropriate broad context.

A number of major problem areas exist at the national, regional, state and local levels which would benefit from the existence of such a systems center. These include power systems, communications systems, national defense systems, inflation, unemployment, energy, pollution, resource management, food, health care systems, transportation systems, urban systems, and industrial productivity. Policy and decision makers in the various sectors of our society need better ways to enhance their policy making abilities.

Models play an important role in the analysis of large scale systems. Short term and long term decisions depend on the notion of a systems model. This may simply be a mental model in the mind of a decision maker or a simulation model stored in a computer. It may be an analytical model, which is a mathematical expression whose solution exhibits characteristics of the system in question. Several thousand models exist today which are designed to help make specific decisions. Attempts to interconnect them in various ad hoc ways to give more global insight generally lead to unreliable results. One consequence of this is that the credibility of models as an aid to decision making suffers. The world is dynamic and stochastic. Unfortunately, most existing models are static and deterministic. Thus the real value of current models is limited. There are competing methodologies which presently exist and which have little connection between them. The typical example is system dynamics models versus econometric models.

The availability of computers has focused attention on the use of models for the analysis and understanding of possible results from various decision alternatives. In this process of using large scale systems models a number of difficulties have been revealed which have limited the effectiveness of these studies. Many of the models are not global enough and show only one area or region and do not provide any feedback or interaction with the rest of the world. To date there are no adequate ways of demonstrating the range of validity of large scale systems models. Many of the models are inherently static, as mentioned before. Databases have generally proven to be inadequate. The data are not timely, they are not up to date, they are of questionable validity, and data formats are incompatible from one model to another. Although different models are nominally related they are generally unable to work together. It is difficult to obtain adequate documentation on how the models are built and how they are intended to be used. Many stochastic effects and uncertainties are simply not incorporated in these models.

INSTITUTIONAL ASPECTS

A government interagency committee for "Large Scale System Analysis" is in the process of formation to address itself to several aspects of large scale systems problems. It is hoped this committee will encourage and guide the evaluation and assessment of existing and planned models and methodologies illuminating those areas most in need of development. The committee plans to recommend research in systems analysis, modelling and decision making to the National Science Foundation and other governmental agencies and organizations. It intends to foster the development and maintenance of a facility for the exploration of policy alternatives. This facility will include a library of simulation models, decision tools, and data banks.

Industrial productivity is another area which certainly could benefit from the development of a national systems center. Problems of coordination, planning and expansion are not well understood. Such issues are affected by technological advances, tax policies, regulation policies, subsidy policies, balance of payments issues, tariffs structures, and others. As mentioned earlier, several complex engineering systems are not well understood. These include power networks, communications networks, for both civilian and military command and control applications, food processing, distribution networks and others.

Universities also have needs which could be filled by a national systems center. Much of the intellectual capital of universities is wasted on irrelevant and trivial research. No mechanism exists for channelling research in directions of meaningful and pressing problems except through very indirect mechanisms of funding agencies such as the National Science Foundation. Generally, mission oriented agencies produce RFP's which are short range in nature. Industrial support for university research has been limited and there seems to be a continuing decline in university/industry cooperation. Information dissemination seems to be fragmented; specialists talk to specialists, publication of new research results is often delayed. There is a lack of appropriate educational materials, not only for university students, but also for the public at large. Universities do not have access to most existing models, either due to a conscious effort on the part of the model builders or simply incompetence in the area of documentation and validation. High quality software for model estimation and optimization generally is not available and not easily transferred.

PROPOSAL FOR A NATIONAL SYSTEMS CENTER

We have seen that a national need exists in the area of complex interacting and interconnected physical, engineering, social and economical systems. The National Systems Center would have a number of major responsibilities which could help alleviate some of these problems. There is a need to evaluate existing theories and describe in detail the need for additional basic and applied research in the area of complex, large scale systems. It could examine limitations of present models of large scale systems and delineate in an objective fashion the areas of applicability. It must identify education and information dissemination needs and provide a user forum to establish a community of interest sufficient to sustain the research and development needed in this field. It could provide the opportunity to cut across artificial barriers to propose directions for systems development. It could focus attention on policy needs and provide the user with the most appropriate tools and best information for policy and decision making.

The National Systems Center would have four basic activities: (1) basic and applied research on large scale systems, (2) basic and applied research on complex models, (3) education and information dissemination, and (4) unique institutional services.

Let us look at the research which may be conducted at a National Systems Center. Such a center could identify and encourage new areas of research with particular emphasis on large scale systems. Furthermore, basic research in model validation which addresses this issue across different modeling philosophies and methodologies could be carried out. This is a task which is greatly needed and is inadequately handled by existing institutions.

The salient characteristics of basic research to be carried out by the National Systems Center include: breadth of scope addressing methodological issues that cut across alternative approaches, a pluralistic attitude which searches for fundamental underlying principles in the study of large scale systems, an exploratory philosophy with the objective of generating interest in research topics both among researchers across the country and in funding organizations, and a focus toward future needs as opposed to review of past problems. Research at the National Systems Center would strive to develop interplay between realistc problems and basic research. These are vital for the development of relevant theories.

It is dangerous to enumerate large scale systems research topics which would be appropriate for NSC. However, let us try to name several as examples of large scale systems research appropriately handled at the center. (1) Credibility of information, distributed data bases, incomplete information, delayed information. (2) Characterization of uncertainty-impact on policy analysis and synthesis. (3) Methodology for cause and effect relations. (4) Identification of systems with changing parameters and structure. (5) Theories of aggregation and decomposition including fast versus slow phenomena and weak dynamic coupling. (6) Organizations under multiple criteria, dynamic teams and games. (7) Conflict resolution theory. (8) Stability of large scale systems, interconnection properties of stable subsystems and global stability. (9) Selection of planning horizons and discount factors. (10) Coordination, centralization and decentralization. (11) Group decision making theories.

Although NSC should undertake both methodological and model development, emphasis should be placed on methodological development, not on the building of models which carry NSC labels. Methodological development may proceed in four areas.

1) Established methodologies including assessment, evaluation, and guidelines for applicability.
2) New methodologies including assessment, evaluation, identification of current and future needs, and formulation of research objectives.
3) Model verification including definitions and procedures.
4) Model validation including research into precisely what constitutes validation and how it can best be carried out.

NSC will need to have access to models in support of methodological research and should be encouraged to take a pluralistic attitude toward modeling approaches. It should provide a forum for creative dissent concerning alternative modeling approaches.

It will be important that NSC maintain models for use by its staff and others and therefore would end up serving as a "foster parent" for models developed elsewhere. This implies that models residing at NSC must be transferable and adequately documented. NSC would provide opportunities for adequate testing and verification and these models will be generally accessible for research. It will be possible to provide some sort of peer review as with most other scientific contributions. A systematic accumulation of experience, development of data on the models, and set of case studies will be undertaken. NSC can serve as the focal point for the development of educational programs in the field of modeling. It can provide the environment for constructive interaction between proponents of alternative methodologies, as well as providing a forum for decision makers and model users and the means for constructive interaction with modelers. It seems that NSC could contribute to increasing the credibility of modeling and models with users and decision makers by evaluating models, providing an independent verification of model results, helping to develop standards for model documentation and developing standards for the documentation of results of studies incorporating models.

Issues relating to modeling research includes model building and testing , model choice and linkage, model verification and validation and model utilization. Under the topic of model building and testing one should look at questions of complexity which relate to how comprehensive large models are constructed, model dynamics which questions the most effective ways for capturing dynamic system behavior, and uncertainty in which problems of objective versus subjective probabilities must be studied. It is important to consider limitations and pitfalls of linking models of the same kind, and also linking models of different types. This is especially true when model development has been carried out for different purposes (e.g., technological forecasting and economic forecasting). One would like to develop operational guides for model verification and validation for both users and modelers. Finally, model use has tended to be rather ad hoc. We must begin to design controlled experiments of model utilization and thus begin to turn large scale systems modeling studies into an experimental science.

INFORMATION DISSEMINATION AND EDUCATION

The unique educational function of the NSC will be carried out at various levels. The NSC will provide the environment for constructive interaction between members of different constituencies such as government, academic and industry, who are all concerned with addressing specific national needs. It will provide a forum for decision makers and model users through which they could address the research community, and also provide a means for constructive interaction with researchers. It can foster constructive interaction between proponents of alternative approaches to large scale system problems and can develop educational programs and aids for government, industry, universities, public consumption and private organizations.

One specific suggestion for educational activity of the NSC is the development and operation of summer institutes. The purpose of these institutes would be to set directions for future research that would contribute to the solution of future mission-oriented problems. This is particularly important because researchers who are establishing tools relevant to future problems are not necessarily aware of what the future problems will be. Policy and decision makers in government and industry can contribute to specifying the future problems, but they are not always aware of the capability of the systems researchers. Although conferences and symposia tend to be helpful in disseminating current results, they generally provide little guidance to future needs.

The summer institutes are envisioned as being annual affairs in which key problems are studied in depth by a selected group of researchers, decision makers and users. They would be multi-week meetings including a staff of core participants and part-time participants. The location of the institute could change from year to year. The idea of an opportunity for in-depth study on both tutorial and research levels in an environment conducive to uninterrupted in-depth discussion is an appealing one and is a format for education and information dissemination not presently used in the systems community.

THE NATIONAL SYSTEMS CENTER

Let us look at the specific organizational concepts developed by the Ad Hoc NSC Committee.

The charter of the NSC would be established as the result of a proposal solicitation effort. The organizations submitting proposals for establishing the NSC would develop its charter as part of their proposal effort. As will be seen later in this paper, the Ad Hoc committee expects that the best institutional arrangement will be a non-profit organization of a somewhat unusual sort. A Board of Governors will be necessary to provide overall policy guidance for the operation of the NSC. This board, made up of eminent systems research specialists from a variety of disciplinary backgrounds, would ultimately be responsible to funding agencies for setting NSC policy and directions. The Director of the NSC will be the key staff member of the entire operation. He should be an individual experienced in the development and management of research with a background which insures that he understands the nature of the research as conducted in various types of institutions in the United States. His offices should be in a small headquarters facility in Washington, D.C. where he will have access to mission-oriented agencies and decision makers who have need for the results of NSC research. At this Washington headquarters there should be a small staff including people responsible for maintaining a library of models and related materials. The Washington staff must maintain close communication with Associate Directors and their staffs and must include liaison people who will maintain contact with policy makers and others.

The Associate Directors of the NSC will be geographically distributed at the various centers of excellence of systems research throughout the country. It is envisioned that large scale projects will be maintained at appropriate geographic centers, such as universities, research institutes and industries where the expertise currently resides or could be conveniently collected. Directors of these large scale projects would be Associate Directors of the NSC and would be phased in and out of NSC operations as their research was phased in and out. These large projects, presumably funded over an extended period of time, would be selected by the NSC Director and Board of Governors in response to proposals submitted by these research groups. Once such a project is funded, the Director of the project becomes an Associate Director of NSC. The committee felt that it would be not only inappropriate but also impossible to try to establish these projects in some remote site such as Washington, D.C.

As has been indicated before, significant systems research projects require extended funding and extended periods of time. It seemed unreasonable to expect that major contributors in the systems field would be prepared to spend extensive periods of time away from their home institution, and therefore this distributed geographical structure is proposed. With this structure in mind, it is seen that the NSC Director and Board of Governors attract the money to NSC and distribute it to these rather large scale projects being conducted in the various geographically distributed centers. The continual process of program evaluation and guidance for future work must eminate from the NSC Director's office. Programs which are recommended by the Director and approved by the Board of Governors should have a mix of funding styles. We envision approximately 50% mission-oriented, 40% funded by the Director to facilitate the technical growth and user-response capability, and 10% of a rather blue sky nature. All of these efforts would be funded in the same way; namely, NSC would obtain outside funding and solicit proposals in the areas of interest. Unsolicited proposals submitted by outside organizations could, of course, also be considered if they fit into the program objectives set by the Director and Board of Governors.

Initially, the NSC should be funded by the U.S. government. It is essential that long-range funding be committed. The National Science Foundation should fund only part of the research effort; the remainder of the funding provided by different government agencies. All funding should be sole source. Industrial and foundation funding should be actively solicited in addition to government funds. The funds necessary for the summer institutes can be provided by the agency or agencies appropriate for the themes of the institute. The NSC is not to engage in competitive bidding. It should not be viewed as being competitive with universities or industry.

SUMMARY JUSTIFICATION

Let us again review why we need a National Systems Center and why the proposed center is better suited to address the needs and problems discuseed when there already exist numerous universities, research institutes, government agencies and industries attacking various subsets of these problems.

Current research in the United States resides in narrowly focused disciplines. Existing institutions in general do not have the appropriate organizational structure, stability of resources, and incentive system to effectively analyze complex dynamic interdisciplinary systems. Government agencies are beginning to recognize the need for a sustained, long-range high quality effort to create necessary interaction among relevant disciplines and large scale systems and to extend current methodologies and develop new ones to address major national problems and ultimate policy issues.

One might ask about existing alternatives to the proposed structure. Government research laboratories carry out research and monitor research in support of the mission of the department with which they are associated. This includes the Department of Defense, Department of Energy, Department of Agriculture, National Aeronautics and Space Administration, etc. They are not chartered to address issues that cut across department boundaries. These are often common issues that are not well understood.

Extramural research is carried out through RFPs that are problem specific with a narrow focus and short range. Time constraints, in such RFPs, often preclude the use of state-of-the-art technology.

The National Academies of Science and Engineering have the ability to bring together for short periods of time, top experts on a wide range of disciplines to make analyses and recommendations on current issues. They often identify research needs, but do not conduct the research. This is quite different than the Academies of Science of other countries. There is nothing to suggest that the American Academies will change this mode of operation.

An individual university, which conducts substantial research activities in a variety of areas, cannot command the resources of the national systems community for a sustained long-term effort. Existing departmental organizational structures do not provide necessary incentives for the faculty to engage in serious long term interdisciplinary research of the type needed for the resolution of large-scale systems problems. Industry also conducts research, but its goal is related to a profit motive. One could argue that there is indeed a profit motive in the solution of large-scale systems problems. Nevertheless it is very difficult to justify a particular company spending the kind of money and effort that would be required to resolve such problems.

Professional technical societies in the systems field rely on volunteers. Committees can be organized to draw upon existing knowledge and make recommendations, but certainly cannot be expected to conduct research and development.

Non-profit organizations, as they presently exist, cannot command the resources of the national systems community for a sustained long-term effort. Moreover, they are not geographically distributed in a way needed for the success of systems research efforts.

Figure 1 at the end of this paper shows the conclusions which must invariably be drawn when examining the existing alternative institutional structures for conducting the work of a National Systems Center. The National Systems Center would essentially be a non-profit organization with an organizational and geographical structure that would permit this type of work to be carried out. It is envisioned by the Ad Hoc NSC Committee that the eventual institutional design for the National Systems Center would indeed be some form of a non-profit organization.

The National Systems Center, as conceived above, would be a unique resource because of the combination in one organization of two key attributes. 1) A primary emphasis on development and documentation of methodologies for decision making and policy analysis and syntheses in those national and regional problem areas where a comprehensive systems description or knowledge is essential. 2) A sustained, stable, and sufficient effort by top researchers and experts in the systems and policy fields cooperating with universities, industry, government and other research organizations to affect a major advance in the state-of-the-art of this field.

The distinctive attributes of the proposed National Systems Center include the accumulation of models and case studies, peer review which will contribute to experimental evidence of data essential for defining methodological research, development of guide lines or standards for model documentation, and availability of advice to decision-makers on the state-of-the-art in large-scale systems studies. Strong interactions with the systems community will ensure that the most appropriate methodology is used in any mission oriented study. At the present no such institutional structure exists in the United States. The National Systems Center will address issues in more depth than the Academies of Sciences or Engineering and with a more holistic flavor than is carried out by existing research institutes, industry and universities. NSC will have a focused basic research component and the research findings will be accessible and widely disseminated.

Thus the Ad Hoc NSC Committee believes there is an urgent need in complex national systems which requires a new organizational structure to carry out basic and applied research in the areas of large-scale systems, modeling, policy analysis and policy synthesis. The proposed National Systems Center represents a unique match between national needs and organizational design.

ACKNOWLEDGEMENTS

It was mentioned at the beginning of this paper that the work that has gone into the development of the proposal contained in the paper is a result of efforts by a large number of people. The following individuals have been the active contributors to the Ad Hoc Committee for a National Systems Center. This group of people was supported in part by National Science Foundation Grant NSF/ENG77-07777 and the nature of the discussion in this paper and the conclusions should be attributed largely to the work of this committee.

Professor Michael Athans, MIT (Chairman)
Dr. A.W. Blackman, U.S. Department of Energy
Dr. S. Brodski, Office of Naval Research
Professor K. Chen, University of Michigan
Dr. H. Chestnut, General Electric Company
Mr. R. Crosby, Department of Transportation
Dr. R. Drenick, National Science Foundation
Dr. J. Fearnsides, Department of Transportation
Mr. L. Fink, Department of Energy
Professor K.S. Fu, Purdue University
Professor J. Gibson, University of Virginia
Professor G. Goodson, Purdue University
Professor L. Kazda, University of Michigan
Dr. R. Larson, Systems Control Incorporated
Dr. A. Levis, Systems Control Incorporated
Professor W. Linvill, Stanford University
Professor J. Palmer, Metropolitan State College, Denver
Professor H. Rigas, Washington State University
Professor T. Saaty, University of Pennsylvania
Professor J. Warfield, University of Virginia
Professor M. Wozny, Renssalear Politechnique Institute
Professor L. Zadeh, University of California, Berkeley
Professor Stephen Kahne, Case Western Reserve University

FIG. 1. SOME EXISTING ALTERNATIVES

	GOV'T. LABS	NAS/ NAE	UNIVERSITIES	PRIVATE INDUSTRY	TECH. SOCIETIES	NON-PROFIT	NSC
CURRENT ORGANIZATIONAL STRUCTURE	NO	NO	YES	NO	NO	NO	YES
STABILITY OF RESOURCES	YES	NO	NO	NO	NO	YES	YES
INCENTIVE SYSTEM	NO	YES	NO	NO	NO	YES	YES

QUANTIZATION LOSS IN OPTICAL COMMUNICATION SYSTEMS*

DONALD L. SNYDER and IAN B. RHODES
Washington University
St. Louis, Missouri 63130

ABSTRACT

In a digital communication system employing an optical carrier and direct detection and having five to forty dark-current counts per channel use, about 1.5db more signal energy is required with hard decisions to achieve the same cutoff rate as with infinitely soft decisions.

This research was supported by the National Science Foundation under Grant ENG 76-11565, by the National Institutes of Health under Research Grant RR00396 from the Division of Research Resources, and by the Office of Naval Research under Contract N00014-76-C-0667.

ASSESSMENT OF CODING POWER GAIN ON A SATELLITE COMMUNICATIONS LINK*

PAUL K. LEE
The MITRE Corporation, Bedford, Massachusetts 01730

ABSTRACT

Coding performance is evaluated in terms of power savings over an uncoded counterpart. The proposed criterion requires maintenance of the same rate of information transfer for all candidate systems. Various convolutional codes employed in the communications system of Lincoln Experimental Satellite numbers 8 and 9 (LES 8/9) are evaluated. Coding power gains based on theoretical and in-orbit link performance are presented.

I. INTRODUCTION

Power for satellite communications between ariborne terminals is valuable because its supply is limited. To assess the potential of error correction codes for conserving system power, satellite system designers need better tools and information. For this purpose, this paper proposes a simple and practical criterion, which is expressed in terms of effective power gain over an uncoded counterpart. This criterion was developed to evaluate the coding performance on a communications link of Lincoln Experimental Satellite numbers 8 and 9 (LES 8/9), a digital processing satellite communications system. [1]

Signal-to-noise density ratio (P_r/N_o) is employed in the criterion for comparing the power requirements between coded and uncoded systems. To avoid favoring low rate codes, comparison is made at a prescribed user bit error rate (BER) of interest. This approach has two advantages: 1) it is simpler and easier to implement than a previous approach, [2] since P_r/N_o is directly measurable in a channel and is often expressed in test results; 2) coding power gain derived from the criterion can be reflected in a direct cutback of the system power budget required by the uncoded counterpart, without decreasing the rate of information transfer.

The LES 8/9 communications link which was evaluated employs various convolutional codes. The codes work in conjunction with differential quadriphase shift keying (DQPSK) with 30-degree decision sectors and differential phase shift keying (DPSK) modulation schemes. A detailed discussion of the coding and modulation schemes as well as the proposed criterion is contained in Section II. This study exploited the unique opportunity to examine the interaction between a code and its interfacing modulation-demodulation (modem) scheme. For the coding assessment, both theoretical and in-orbit link performance results are included in Section III. Conclusions of the study are presented in Section IV.

---

* This work was supported by the Electronic Systems Division of the Air Force Systems Command.

II. CODING AND MODULATION SCHEMES ANALYZED [1]

The coding and modulation schemes for evaluation were employed on a communications link of LES 8/9.

The LES 8/9 communications system consists of two near synchronous altitude spacecraft, designated as LES 8 and LES 9, which were launched on 14 March 1976 from Cape Canaveral to demonstrate advanced satellite communications technology. The LES 8/9 system has three types of terminals. Together with signal processors on board the satellites these terminals provide a forward link, a report back link, and a conference link. In addition to the three communications links, a DPSK modulated crosslink serves as a digital pipe between the satellites and extends the communications coverage. The coding and modulation schemes under evaluation were employed on the conference link.

The conference link provides the means to pass record, data, or vocoded voice communications among airborne terminals. Available user data rates include a low rate of 600, 1200, or 2400 bits per second (bps), as well as a high rate of 4800, 9600, or 19,200 bps. The signal is modulated in DQPSK and frequency hopped every 5 msec for the K_a-band (36 to 38 GHz) uplink transmission. Onboard the satellite, the uplink DQPSK signal is demodulated and remodulated into DPSK for K_a-band downlink or optional crosslink transmission.

The channel rate of the conference link is fixed at 10 kbps or 100 kbps[*] depending upon the actual user data rate. At the low data rate, there is one frame to the 5 msec hop interval and there are 25 DQPSK symbols to the frame. (Each symbol carries two bits of channel information.) Only 18 symbols contain conference data. The other symbols, which are overhead in nature, are employed for assisting signal acquisition or probing the channel.

The uplink DQPSK signal is demodulated at the satellites with angular quantization. In this approach, the phase of the received signal is quantized to 12 equally spaced values. Each sector corresponds to 30 degrees. On the other hand, the DPSK downlink signal is processed by the conventional Costas-loop phase-lock DPSK demodulator.

The conference link handles three different data rates, 19.2, 9.6, and 4.8 kbps when the channel rate is 100 kbps, and 2.4, 1.2, 0.6 kbps at the 10 kbps channel rate. The transmission of different user data rates through fixed channel rates is accomplished by allowing a variable code rate. That is, the encoding is designed to operate at R = 1/3, 1/6, or 1/12. The variable code rates are obtained by concatenation. There is an inner convolutional code with constraint length 7 that can have a code rate of either 1/3 or 1/6, and an outer convolutional code with constraint length 10 and rate 1/2. By selecting the inner code rate and the option of using the outer code, one can have a code with rate 1/3, 1/6, or 1/12.

The inner code is decoded with a Viterbi algorithm operating at R = 2/6 or R = 2/12. The Viterbi algorithm is 2/6 and 2/12 instead of 1/3 and 1/6 because the data are transmitted in pairs as DQPSK and are therefore processed in pairs as DQPSK by the decoder. The outer code is decoded with a K = 10, look ahead L = 11 feedback decoder, and this decoder operates on a binary symmetric channel.

---

[*] These rates apply only to a single satellite configuration. If uplink and crosslink data are multiplexed, each rate will be doubled.

Interleaving is processed before the bit stream is DQPSK modulated for K_a-band uplink transmission. Deinterleaving is processed before the demodulation of the DQPSK signal at the K_a-band downlink terminals. The coding and modulation characteristics of the conference link are summarized in Table 1.

Table 1

Coding and Modulation Characteristics

User-to-User Data Rate	600/1200/2400/4800/9600/19,200 bps
Source-Data Encoding	Rate - 1/3, 1/6, 1/3 x 1/2, or 1/6 x 1/2 Convolutional with interleaving. $K^* = 7$ for rates 1/3, 1/6, K = 10 for rate 1/2.
Uplink Modulation	DQPSK, 30° decision sectors
Uplink Frequencies	K_a-band (36 to 38 GHz)
Satellite Baseband Converter	DQPSK
Downlink Modulation	Binary DPSK
Downlink Frequencies	K_a-band (36 to 38 GHz)
Output-Data Decoding	Rates 1/3, 1/6, hard-decision Viterbi decoding. Rate 1/2, feedback decoding. (Decoding after de-interleaving)

*Constraint length.

III. ASSESSED CODING POWER GAINS

It is advantageous to employ P_r/N_o as a system parameter for evaluating the coding performance. Employment of high redundant codes will generally result in either an increased modulation rate* in the channel or a decrease in the rate of information transfer. To avoid the possibility of favoring high redundance codes, the criterion requires that an equal rate of system information transfer be maintained for all candidate systems. The coding equipment therefore can be viewed as a black box to the system user. The merit to be judged is its influence over the system power budget at a user BER of interest. Actually, maintaining an equal rate of system information transfer is the first condition to meet in the definition of coding power gain employed in this report. With the first condition met, the coding power gain, denoted by G_c, is defined as the algebraic difference obtained by subtracting the coded signal-to-noise density ratio, $(P_r/N_o)_c$ in dB, from that of the uncoded counterpart, $(P_r/N_o)_u$ in dB. The subtraction is taken at a specified user BER.

---

*An increase in modulation rate means also an expansion of channel bandwidth, which may trade for a better modem design. This, however, is not pursued further in this paper.

Mathematically, the coding power gain can be expressed as:

$$G_c = \left(P_r/N_o\right)_u - \left(P_r/N_o\right)_c \quad \text{dB} \tag{1}$$

with an equal rate of information transfer, and at a specified user bit error rate.

The coding and modulation schemes in LES 8/9 were designed to allow for measurements of channel bit error rate for uncoded performance, and user bit error rate for coded performance. This approach has an apparent advantage in that only one modulation rate is required to measure necessary coded and uncoded performances. However, the test results would not be based on the same rate of information transfer. In the calculation of coding power gain, proper adjustment of the test results has been made to assure an equal rate of information transfer between the coded and uncoded links.

A full demodulation and remodulation is carried out on-board the LES 8/9 spacecraft between each up-, cross-, and downlink component. This permits independent BER measurements on one component link (uplink or downlink) if the P_r/N_o of the other component link is maintained sufficiently high. The link data reported in the following were taken in a benign, fade-free environment.

Coding Gain on DQPSK Modulated Uplink

The BER curves for the uplink uncoded and coded performance are presented in Figures 1 and 2, respectively. Without losing generality, channel rate at 100 kbps is considered. In this mode, the information rate at which a coded channel can transfer is equal to the channel rate (100 kbps) multiplied by the code rate, R. The resulting user rate is higher than that actually delivered to the user. This is due to the overhead bits employed.

In deriving the theoretical coding gain, theoretical DQPSK performance and coding performance predictions were employed. The theoretical performance for the coding-decoding schemes can be analytically expressed as:

R = 1/3

$$P_o = 389000 \left(P_i\right)^{8.215} \tag{2}$$

R = 1/6

$$P_o = 10^{9.8} \left(P_i\right)^{17.4} \tag{3}$$

R = 1/3 x 1/2

$$P_o = 10^{26} \left(P_i\right)^{32.45} \tag{4}$$

R = 1/6 x 1/2

$$P_o = 10^{56} \left(P_i\right)^{88} \tag{5}$$

In the above expressions, P_o denotes the BER at the decoder output, and P_i the channel BER input to the decoder.

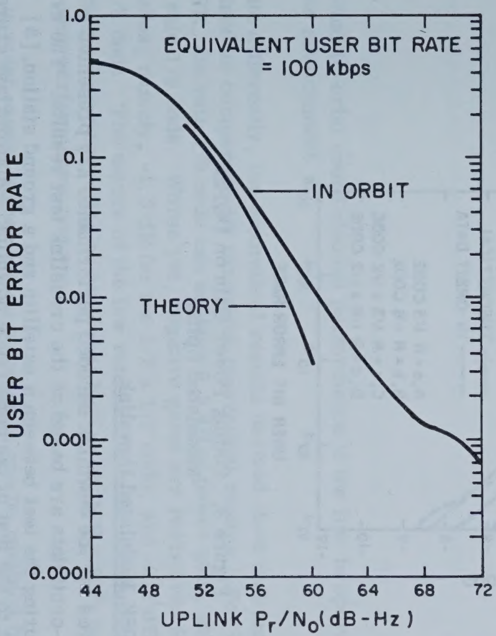


Figure 1. Uncoded DQPSK Modulated Uplink Performance (From Ref. 1)

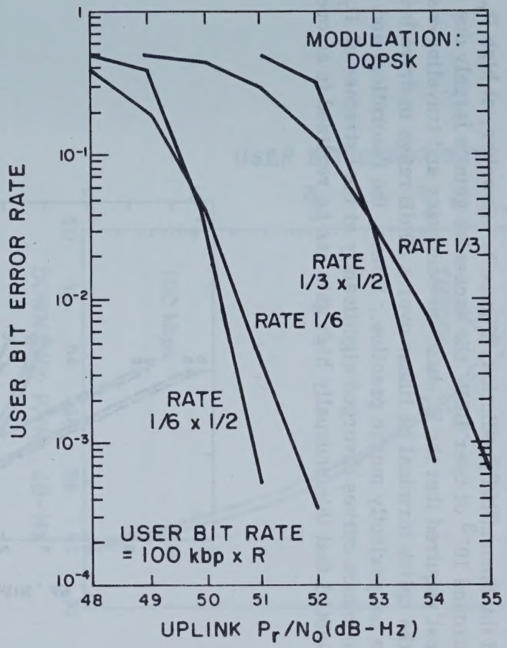


Figure 2. Measured Coded Uplink Performance (From Ref. 1)

The theoretical results show that coding gains rapidly as the user BER decreases. Examples of the coding power gain are: for theoretical results of the uplink performance at 10^{-3} user BER, about 3 dB for the rates 1/3 and 1/6 codes, and 1.9 dB for the rate 1/3x1/2 code. For the same uplink, the in-orbit test results show a much improved performance, as seen in Figure 3. The improvement may be partially attributed to the fact that the in-orbit DQPSK performance curve is not as steep as that of the theoretical prediction, Figure 1. However, at high P_r/N_o, corresponding to around 10^{-3} of user BER, the excessive gain is largely due to the test setup. The test required that the K_a-band up-converter and traveling-wave-tube amplifier in the uplink terminal be linear over a 30 dB range in drive levels.[1] This condition was not perfectly met in practice. Hence, the theoretical and in-orbit link performance curves diverged significantly at the extremes of P_r/N_o, Figure 1. It is doubtful that this unusually high gain can be realized in an operational system.

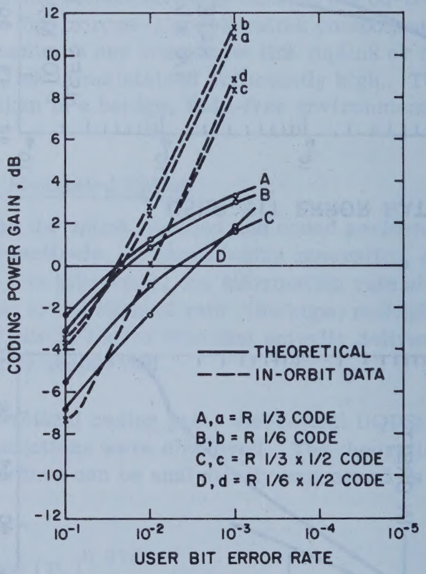


Figure 3. Coding Power Gain on DQPSK Modulated Uplink

Coding Gain on DPSK Modulated Downlink

The BER curves for the downlink uncoded performance are presented in Figure 4. The in-orbit data are based on the crosslink test results. The crosslink was established during the test between a satellite and a ground station.[3] This setup is equivalent to the test of the conference downlink. Appropriate expressions contained in Eqs. (2) to (5), as well as the assumption of 100 kbps channel or data rate, are implemented to assess coding gain. The coding gain derived from the in-orbit uncoded DPSK channel performance and the theoretical decoder performance given by Eqs. (2) to (5) is called "empirical coding power gain." It is expected that the empirical gain would not differ significantly from the gain derived

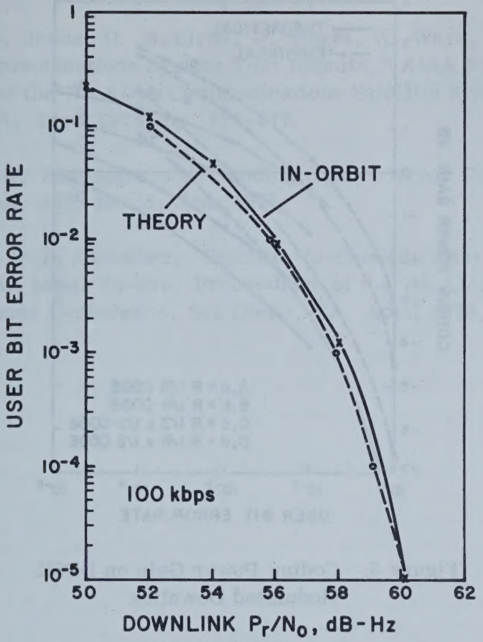


Figure 4. Uncoded DPSK Modulated Downlink (In-Orbit Data From Ref. 3)

from actual in-orbit coded channel performance if the link is operated in a benign, fade-free environment.

Quite differently, the theoretical results derived show that much less gain can be obtained as compared with the uplink theoretical results. For instance, at 10^{-3} user BER, the rate 1/3 code can achieve 0.8 dB of power gain, and almost no gain for the rate 1/6 code. Worse yet, negative gains are registered for the concatenated codes, namely, -1.3 dB for the 1/3 x 1/2 code, and -2.2 dB for the rate 1/6 x 1/2 code. The nature of the low rate of the codes together with a steeper modulation scheme operating at somewhat high user BER (10^{-3}) largely accounts for the poor performance outcome.

As in most cases, higher coding power gain can be realized from the in-orbit channel performance data. Even when the in-orbit DPSK channel performs very closely to the theoretical prediction, it offers no exception. This point, along with other results, can be found in Figure 5.

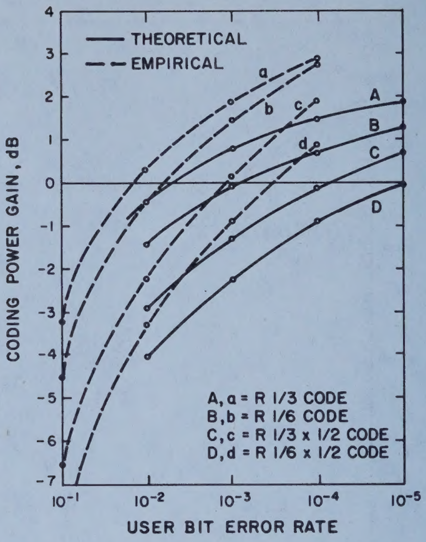


Figure 5. Coding Power Gain on DPSK Modulated Downlink

IV. CONCLUSIONS

Coding performance on a LES 8/9 communications link operated in a benign, fade-free environment has been evaluated in terms of power gain, a criterion especially desirable for system analyses and applications. With the unique combinations of coding and modulation on the LES 8/9 link, excellent opportunities have been made available to examine the interaction between a code and its interfacing modulation-demodulation scheme.

For a given code and its associated decoding algorithm, it was noted that the coding power gain is largely determined by its modulation-demodulation (modem) scheme. As the performance curve (BER vs. P_r/N_o) of a modem becomes steeper, the effective power gain is reduced.

It was also noted that a threshold point in the coding performance exists. When a system operates at a user BER higher than the threshold point, coding deteriorates into a negative gain region, which by all standards is undesirable. As the user BER drops lower than the threshold point, coding starts to gain positively. The behavior of the threshold point was studied by exploiting the convolutional code, which offers selectable rates. For a given modem scheme, it was noted that the threshold point moves toward low user BER as the rate of a code decreases. Also, the rate of effective power gain increases more rapidly. This suggests that a low rate code which may achieve high power gain at very low user BER, e.g., 10^{-5} or less, may not be efficient at a moderate or high user BER, e.g., 10^{-3} or more.

Consequently, the desirability of a code depends not only on the interfacing modem, but also on the range of user BER over which the system is intended to operate.

REFERENCES

1. L. Collins, L. Jones, D. McElroy, D. Siegel, W. Ward, D. Willim, "LES 8/9 Communications System Test Results," AIAA Paper 78-599, Proceedings of the 7th AIAA Communications Satellite Systems Conference, San Diego, CA, April 1978, pp. 471-477.

2. J.H. Blythe, K. Edgcombe, "Net Coding Gain of Error Correcting Codes," Proc. IEE, Vol. 122, No. 6, June 1975.

3. D.M. Snider, D.B. Coomber, "Satellite-to-Satellite Data Transfer and Control," AIAA paper 78-596, Proceedings of the 7th AIAA Communications Satellite Systems Conference, San Diego, CA, April 1978, pp. 457-470.

VITERBI DECODER SYNCHRONIZATION AND BER MEASUREMENT USING METRIC VALUES

RICHARD W. KORALEK
San Jose State University
San Jose, CA

MALCOLM A. METZLER
Ford Aerospace & Comm. Corp.
Palo Alto, CA

ABSTRACT

It is shown by computer simulation that the distribution of metric values in a Viterbi decoder depends on the E_b/N_o of the communications link. A particular measure of the metric distribution is developed which is very easy to compute in hardware. It is shown that this measure is an accurate and fast indication of synchronization down to very low values of E_b/N_o and that it also gives an indication of average bit error rate on the link.

INTRODUCTION

Many modern digital communications systems include a forward error correcting (FEC) system. Use of an FEC reduces the required signal-to-noise ratio to achieve a given bit error rate and can thereby significantly reduce system cost. A very common FEC system consists of a convolutional encoder at the transmitter and a maximum likelihood decoder using the Viterbi algorithm (1) at the receiver. Such a system has an extremely simple encoder but a rather complex decoder, making it ideal for satellite-to-ground links. The decoder typically operates on soft receiver decisions to improve gain.

A convolutional encoder is shown in Figure 1. For a rate K/n convolutional code, the encoder accepts data in blocks of K bits. Each such block is encoded into an n-bit block of code bits. The constraint length of the code, K, is the length of the encoder shift register. At the decoder code bits are accepted in blocks of n bits. The decoder calculates the most likely set of original information bits that would result in the received sequence of code bits. In order to decode properly, the decoder must know the boundaries of the n-bit code blocks. If these boundaries are incorrect, the decoder will misinterpret the code bits and will not be able to properly reconstruct the original data bits. Finding the proper boundaries of the n-bit code blocks is referred to as node synchronization.

One possible way of attaining node sync is to keep track of the approximate decoded error rate, using a system such as that shown in Figure 2. The decoder output, which is assumed to be nearly correct information bits, is re-encoded. The encoder output should now correspond to the original received code bits, and the number of disagreements is counted. As long as the decoded error rate is small compared with the received channel error rate, this count gives a good indication of the channel error rate. If the decoder is out of sync, however, the error rate indication will be approximately 50%. This very high error rate can then be used to indicate an out-of-sync condition.

There are several disadvantages to this method. One is that a fairly long stream of data must be examined to achieve a sufficiently accurate error rate measurement to reliably indicate sync. We have found experimentally that, under certain channel conditions, an out-of-sync condition may yield a measured error rate as low as 25%. In such a case, a fairly long measurement must be made to distinguish this from a noisy channel in-sync condition with an error rate of, say, 5%. To compound this problem, Viterbi decoders produce errors in bursts (2) so the error rate measurement must be long enough to swamp out any burst error effects.

Since the system shown in Figure 2 does measure decoded bit error rate (BER), it could be used to give an indication of link E_b/N_o. Such an indication could be used, for example, for adaptive network control. The problem is that very long measurement times are necessary. The system actually counts errors as they occur, and at the very low error rates typical of operating systems (typically 10^{-6} and lower), a very large number of bits must be examined before a statistically reliable BER estimate can be made.

The remainder of this paper describes a new method of detecting synchronization and measuring average link BER using the values of the metrics used in the decoding process. The advantage of this method over other node synchronization methods is that it can be accomplished in a very short time with high accuracy. It also gives a very fast and reliable indication of BER, which other methods cannot do. In addition, this method makes use of the data already in the decoder and therefore uses very little additional hardware.

DESCRIPTION OF THE VITERBI ALGORITHM

A typical convolutional encoder is shown in Figure 3. This particular encoder has a constraint length K=3 and a rate of 1/2. The encoder can be thought of as transitioning through a sequence of states as data is shifted through it. The state of the encoder is given by the leftmost K-1 bits of the encoder shift register. The sequence of possible states that this encoder can go through is shown in the trellis diagram of Figure 4. In this diagram, a solid line represents a state transition when the data bit is a 0, and a dotted line represents the state transition corresponding to a 1. Thus, a sequence of data bits will cause the encoder to trace out a particular path through the trellis diagram.

The job of the decoder is to determine the most likely sequence of encoder states, or path through the trellis, given the received data. To do this, the decoder assigns a figure of merit, called a metric, to each possible path. This metric value is a measure of the agreement between the correct code bits for a given path and the actual received code bits. For example, if a 3-bit receiver soft decision ranges from 000 for a noiseless received 0 to 111 for a 1, then the metric value would consist of the 3-bit soft decision if the correct code bit should be a 1 and the complement of the 3-bit soft decision if the bit should be 0. Thus the maximum metric corresponds to best agreement. The metric value for a given path is the sum of all the individual metrics for the bits that make up the path in the trellis diagram. After a few steps, the number of possible paths through the trellis grows excessively, so the Viterbi algorithm eliminates half the paths at each step. This is done by comparing the metric values of the paths leading into any given node in the trellis and retaining only the path with the highest metric. In this way, the number of paths that must be traced remains at 2^{K-1}. The method is still maximum likelihood because each of the discarded paths is less likely than the other path into that node.(3)

METRIC DISTRIBUTIONS AND BER

A typical distribution of the metric values of the various paths at any given decoding step is as shown in Figure 5. This figure shows the average distribution of metrics for a K=7 rate 1/2 code for several values of E_b/N_o. These values, as well as the values shown in the rest of this paper, were derived by computer simulation of a Viterbi decoder and were verified in actual operating hardware decoders. Note that the shape of the distribution depends quite strongly on the value of E_b/N_o. At higher values of E_b/N_o, the metric distribution is relatively wide, while at lower values of E_b/N_o it narrows considerably. This occurs because at low E_b/N_o (high channel error rate), even the correct path has many disagreements with the received sequence while many incorrect paths may have many agreements.

There are a number of ways we might go about getting a measure of this distribution in order to estimate E_b/N_o, or BER. For example, we might count how many of the metrics are within some value of the highest metric. In this paper, we will use the difference between the highest metric and the lowest metric as a measure of the distribution. This measure is easy to compute in hardware and gives good results.

The difference, Δ , between the highest metric and the lowest metric at any step is bounded by (K-1) M, (3)(4) where M is the maximum value of the incremental metric associated with traversing a branch in the trellis. However, in general, Δ is considerably less than this bound. Figure 6 shows a plot of $\overline{\Delta}$, the average value of Δ , as a function of E_b/N_o for a K=7 rate 1/2 code. Similar data are plotted in Figure 7 for a K=6 rate 1/2 code. It is easily seen that there is a strong linear relationship between E_b/N_o and $\overline{\Delta}$. Thus, measuring the average value of Δ gives a good indication of E_b/N_o (and thereby BER). Figures 6 and 7 also show the standard deviation of Δ.

From these data, it can be seen that an accurate estimate of $\overline{\Delta}$ can be made in on the order of a few tens of steps. Note that no actual decoding errors need to be made in order to estimate $\overline{\Delta}$ and therefore BER. This is an important property, since many other BER measurement techniques must wait for a number of errors to occur. At the very low error rates typical of practical systems, this can take a very long time. However, the measurement of $\bar{\Delta}$ can be made in a few bits.

SYNCHRONIZATION USING METRIC DISTRIBUTION

Figure 8 shows another plot of $\bar{\Delta}$ vs E_b/N_o for the K=7 code, this time for two out-of-sync conditions. The value of $\bar{\Delta}$ is now practically independent of E_b/N_o and remains at a fairly low level. This is because, in the out-of-sync condition, the presumed received bit sequence does not consistently agree or disagree with any of the possible path sequences. Comparing Figure 6 with Figure 8, we see that, using $\bar{\Delta}$, the out-of-sync condition can be differentiated from the in-sync condition at an E_b/N_o as low as about 2 dB. This value can vary depending on how long an average is taken to compute $\bar{\Delta}$. If the number of bits over which Δ is averaged is increased, the estimate of $\bar{\Delta}$ is more accurate. This means either that the false alarm rate will be lower at a given E_b/N_o or that the system can be operated at a lower E_b/N_o for a given false alarm rate. Figures 7 and 9a show similar results for the K=6 code.

Typical FEC systems use some form of PSK modulation technique for transmission. Because of this, the demodulated received data can be the complement of the transmitted data. Many codes, such as the K=7 rate 1/2 code used in our previous examples, are "transparent". For these codes, inverting the coded data on the channel simply inverts the decoded data, so that simple differential encoding and decoding will remove the ambiguity. For other codes which are non-transparent, such as the K=6 code used in our examples, inverting the code bits results in unknown effects at the decoder output, so differential encoding will not work. Thus, for non-transparent codes, whether the bits are inverted or not must be known as well as node sync.

Figure 9b shows values of $\bar{\Delta}$ for the K=6 code when the bits are inverted but in-sync, and Figure 9c shows the data for the case of inversion plus an out-of-sync condition. The data are similar to the case of out-of-sync only. Thus, measuring $\bar{\Delta}$ will differentiate between the in-sync, non-inverted case and the other three cases. However, it is clearly very difficult to differentiate out-of-sync from inversion using a measurement of $\bar{\Delta}$. For this type of code, a low value of $\bar{\Delta}$ must initiate a search algorithm, trying various combinations of inversion and sync and re-measuring $\bar{\Delta}$ until an appropriately high value of $\bar{\Delta}$ indicates that the correct combination has been found.

CONCLUSIONS

It has been shown that the distribution of metric values in a Viterbi decoder gives an indication of synchronization and an indication of average link bit error rate. The method has the advantage that no errors need to be made, resulting in very fast sync and BER indications. It also uses very little hardware beyond what is already present in a Viterbi decoder.

Further analysis is being done concerning the nature of the filtering used to calculate $\overline{\Delta}$ from a series of values of Δ. A simple digital low-pass filter, as shown in Figure 10, appears to work well. By varying the value of α, the averaging length for $\overline{\Delta}$ can be varied. This allows trade-offs between speed of sync indication, probability of false out-of-sync indication, and probability of false in-sync indication. Another variable is the criterion for deciding that an out-of-sync or in-sync condition exists, including the threshold value of $\overline{\Delta}$. It appears that an adaptive system, varying both the threshold and the filter time constant, might be appropriate.

REFERENCES

1. A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm" IEEE Trans. Inform. Theory, Vol. IT-13, pp. 260-269, Apr. 1967

2. J. A. Heller, I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communication" IEEE Tran. Comm. Tech, Vol COM-19, pp. 835-848, Oct. 1971

3. G. C. Clark, Jr., "Implementation of maximum likelihood decoders for convolutional codes" International Telemetering Conference, Sept. 27-29, 1971, Washington, D.C.

4. I. M. Jacobs, "Practical Applications of Coding" IEEE Trans. Inform. Theory, Vol. IT-20, pp. 305-310, May 1974

A RATE $\frac{K}{N}$ CONVOLUTIONAL ENCODER

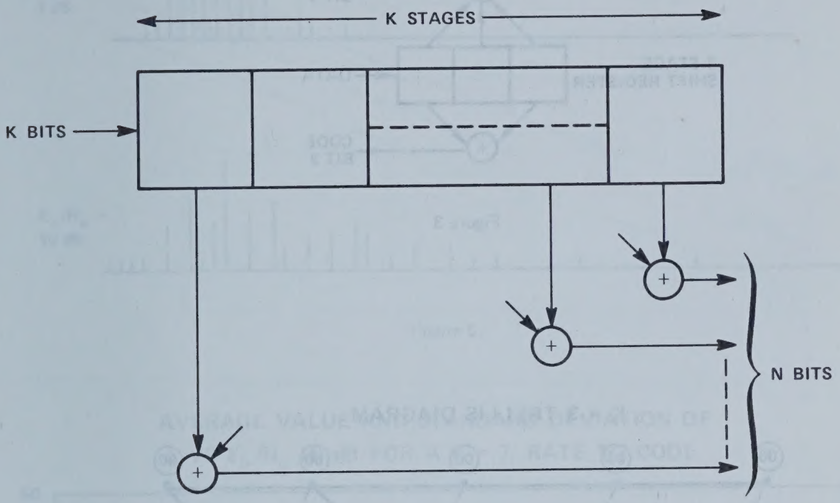

K STAGES
K BITS
+
+
+
N BITS

Figure 1

A CONVENTIONAL NODE SYNCHRONIZATION DETECTOR

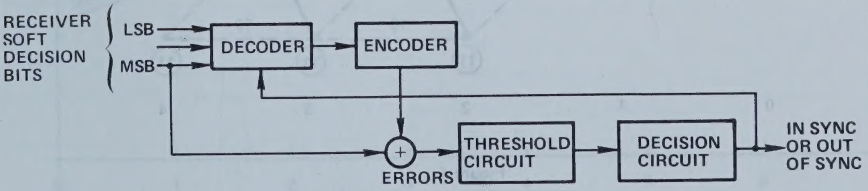

RECEIVER SOFT DECISION BITS
LSB
MSB
DECODER
ENCODER
+
ERRORS
THRESHOLD CIRCUIT
DECISION CIRCUIT
IN SYNC OR OUT OF SYNC

Figure 2

CONVOLUTIONAL ENCODER, K = 3 RATE 1/2

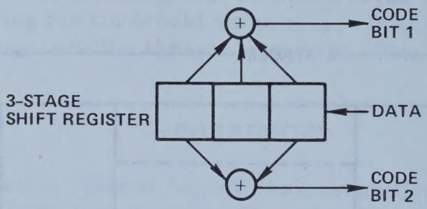

CODE
BIT 1
3-STAGE
SHIFT REGISTER
DATA
CODE
BIT 2

Figure 3

K = 3 TRELLIS DIAGRAM

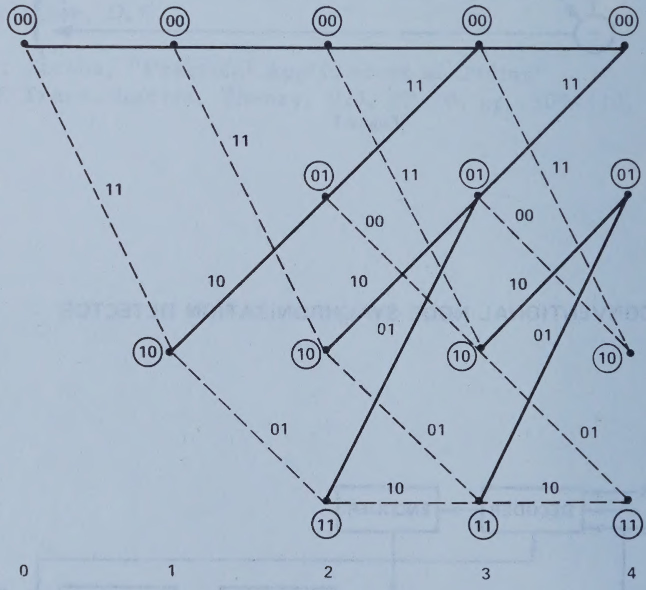

00
00
00
00
00
11
11
11
11
11
11
01
01
01
00
00
10
10
10
01
01
10
10
10
10
01
01
01
10
10
11
11
11
0
1
2
3
4

Figure 4

TYPICAL METRIC VALUE DISTRIBUTIONS

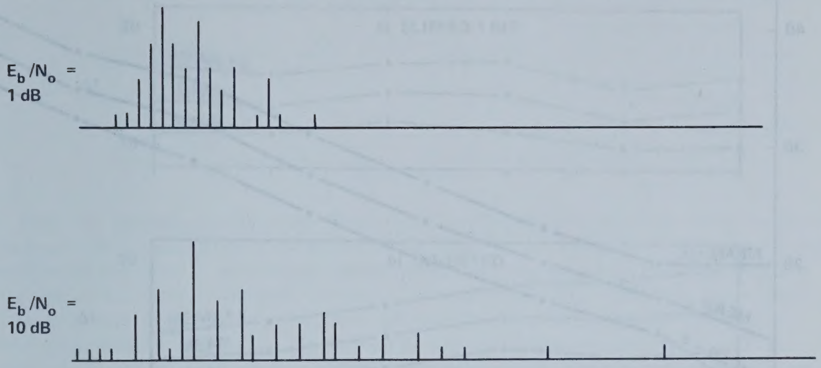


Figure 5

AVERAGE VALUE AND STANDARD DEVIATION OF Δ VS E_b/N_o IN dB FOR A K = 7, RATE 1/2 CODE

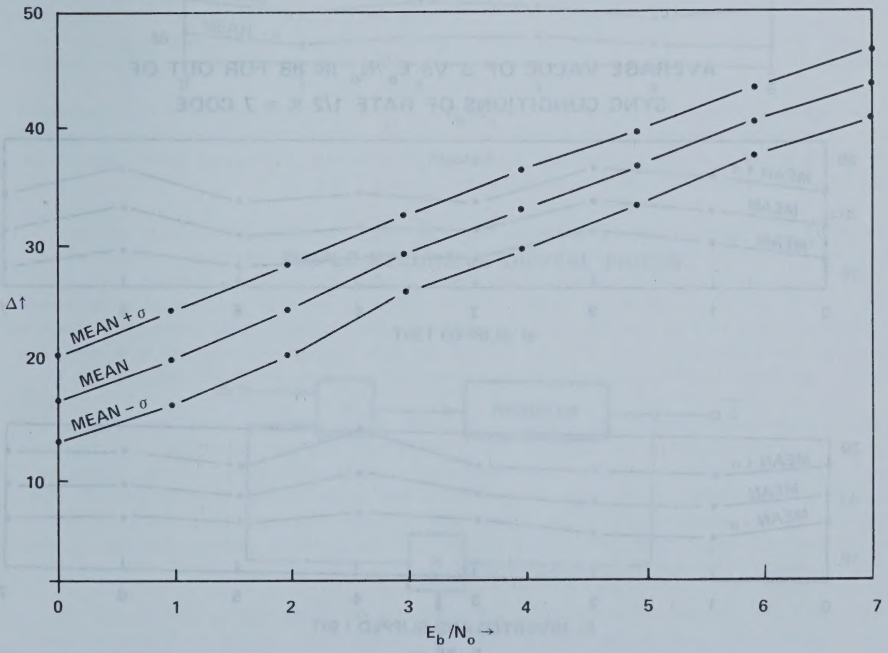


Figure 6

AVERAGE VALUE AND STANDARD DEVIATION OF Δ VS E_b/N_o IN dB FOR A K = 6, RATE 1/2 CODE

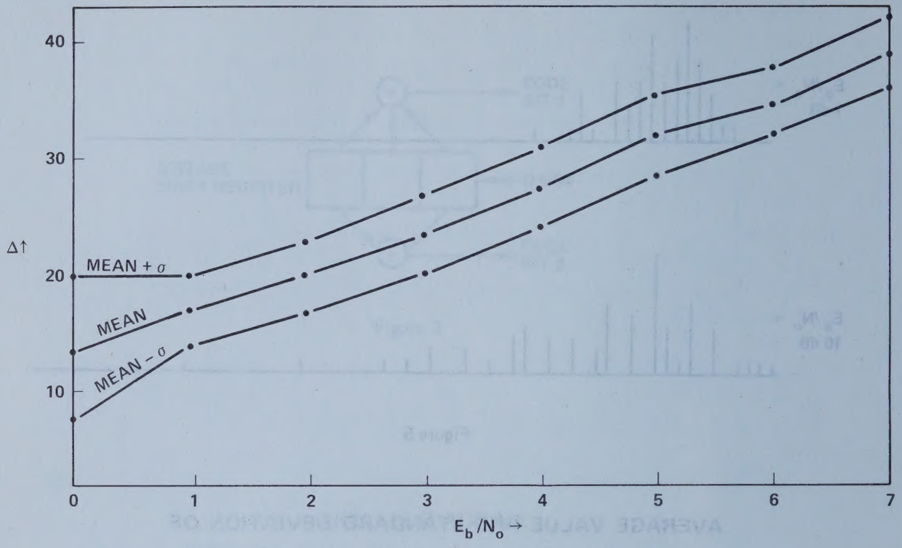


Figure 7

AVERAGE VALUE OF Δ VS E_b/N_o IN dB FOR OUT OF SYNC CONDITIONS OF RATE 1/2 K = 7 CODE

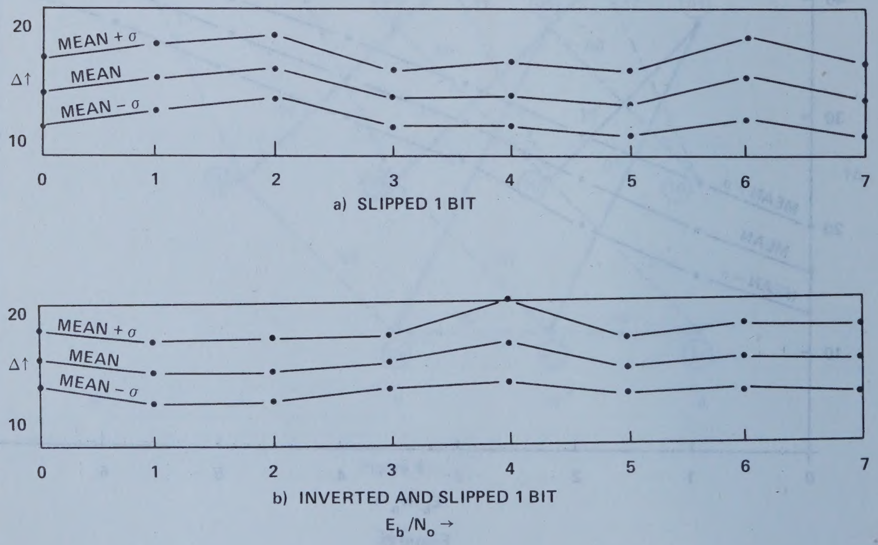


Figure 8

AVERAGE VALUE AND STANDARD DEVIATION OF Δ VS E_b/N_o IN dB FOR A K = 6, RATE 1/2 CODE

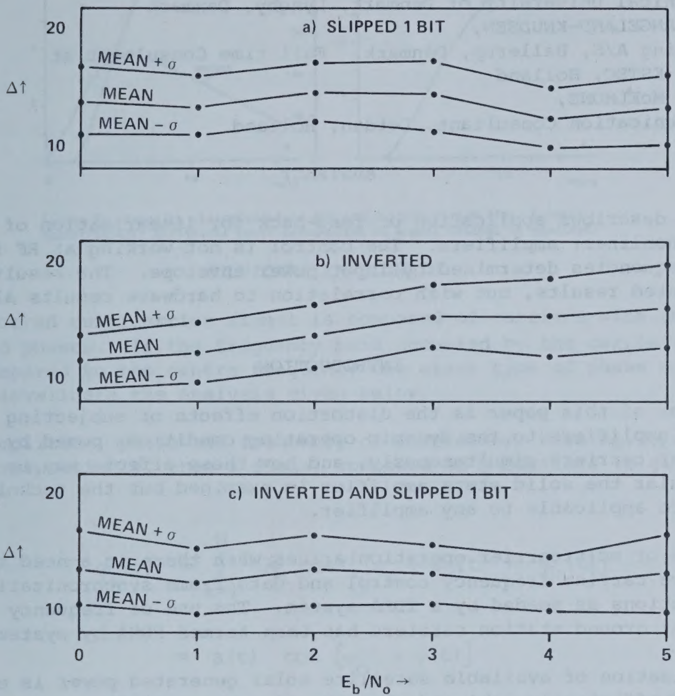


Figure 9

SIMPLE RECURSIVE DIGITAL FILTER

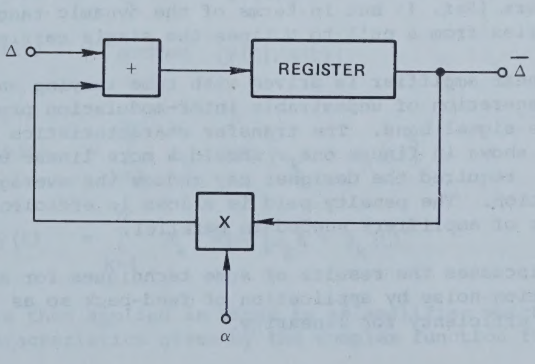


Figure 10

CONTROL THEORY APPLIED TO THE PROBLEM OF REDUCING INTERMODULATION DISTORTION IN NONLINEAR AMPLIFIERS UNDER MULTICARRIER OPERATION

M. BLANKE,
Technical University of Denmark, Lyngby, Denmark
J. LANGELAND-KNUDSEN,
Rovsing A/S, Ballerup, Denmark. Full time Consultant at
ESA, ESTEC, Holland
T.P. McELHONE,
Communication Consultant, Leiden, Holland

ABSTRACT

The paper describes application of feed-back for linearisation of gain and phase of nonlinear amplifiers. The control is not working at RF frequencies but at frequencies determined by input power envelope. The results given are simulated results, but with correlation to hardware results also given.

INTRODUCTION

The subject of this paper is the distortion effects of subjecting nonlinear RF amplifiers to the dynamic operating conditions posed by using independent carriers simultaneously, and how these effects may be improved. In particular the solid state amplifier is examined but the techniques studied are applicable to any amplifier.

The choice of multicarrier operation arises when there is a need to avoid restrictive carrier frequency control and data frame synchronization among ground stations as needed by a TDMA system. The use of frequency independent ground station carriers has been termed FDMA by system engineers.

Full utilization of available satellite solar generated power is economic and is most efficiently achieved when the RF power amplifiers are operated in saturation. This condition is a nonlinear one but even more important it can not be sustained in a multicarrier operation.

The use of independent frequency carriers with random phase and phase modulation generates a composite signal whose power envelop varies with time. The probability of a given power level at a given moment in time has a probability density (PDF) which depends on the number of carriers. Probability density functions of multicarrier signal have been analyzed by earlier writers (Ref. 1) but in terms of the dynamic range of a N-carrier signal it varies from a null to N times the single carrier amplitude.

When a nonlinear amplifier is driven with time varying envelope signal there is a generation of undesirable inter-modulation products falling back into the signal band. The transfer characteristics of one type of amplifier is shown in figure one. Should a more linear operation of the amplifier be required the designer may reduce the average input drive below saturation. The penalty paid is a loss in efficiency and an increase in the number of amplifiers needed in parallel.

This paper discusses the results of some techniques for a reduction of inter-modulation noise by application of feed-back so as to improve the trade off of efficiency for linearity.

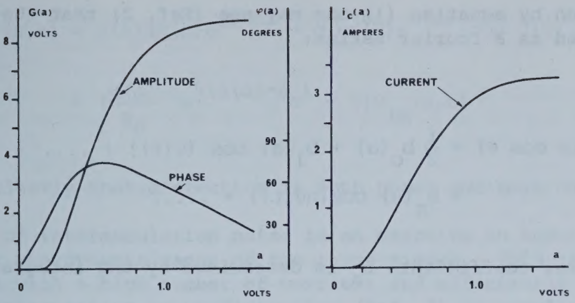


Figure 2. Transfer characteristics for solid-state transmitter. Current characteristic determines power consumption as supply voltage is constant.

MULTICARRIER SIGNALS

An undistorted multicarrier signal is composed of carriers with uncorrelated phases. If the frequency band occupied by the carriers is narrow compared to the centre frequency the exact type of phase modulation will not invalidate the analysis given below.

Let the amplitude, phase and frequency of carrier nr. k be A_k, ϕ_k and $\omega_o + \omega_k$, respectively, then a signal composed of N carriers is given by

$$\begin{aligned} e_i(t) &= \sum_{k=1}^{N} A_k \cos\left[(\omega_o + \omega_k)t + \phi_k(t)\right] \\ &= a(t) \cos\left[\omega_o t + \psi(t)\right] \\ &= a(t) \cos\theta(t) \qquad (1) \end{aligned}$$

where

$$a(t) = \left[x^2(t) + y^2(t)\right]^{\frac{1}{2}} \qquad (2)$$

$$\psi(t) = \arctan\left[y(t)/x(t)\right] \qquad (3)$$

and

$$x(t) = \sum_{k=1}^{N} A_k \cos\left[\omega_k t + \phi_k(t)\right]$$

$$y(t) = \sum_{k=1}^{N} A_k \sin\left[\omega_k t + \phi_k(t)\right]$$

The signal $e_i(t)$ is then applied as input to an amplifier which is described by memory less characteristics given by the complex function $f(e_i(t))$.

As $e_1(t))$ is given by equation (1) one may see (Ref. 2) that the output can be represented as a fourier series:

$$f(a \cos \theta) = \frac{1}{2} b_o(a) + b_1(a) \cos(\theta(t)) + \ldots. + b_n(a) \cos(n\theta(t)) + \ldots.. \tag{4}$$

Each of the fourier coefficients b_n is determined by the integral

$$b_n(a) = \int_0^{\pi/\omega_o} f(a \cos(\omega_o\tau)) \cos(n\omega_o\tau)\omega_o \frac{2}{\pi} \cdot d\tau \tag{5}$$

Note that in the above derivation no restriction has been made on the choice of ω_o. It can then conveniently be chosen as the centre frequency of the band occupied by the multicarrier signal.

Because of filtering, the RF transmission is completely characterised by the fourier component $b_1(a)$ which is independent of the centre frequency, ω_o. In addition, $b_1(a)$ is easily measurable. The RF transfer characteristics given in fig. 1. are actually the components of $b_1(a)$. Fig. 1 stresses the fact that $b_1(a)$ may be a complex valued function; i.e. an input amplitude, a, gives a certain output amplitude of the RF signal, but also a phase shift of the signal. The nonlinear RF transfer can thus be written

$$b_1(a) = a \cdot g(a)\, e^{j\phi(a)} \tag{6}$$

where g(a) is amplitude gain and $\phi(a)$ phase characteristics.

The transmitter response is also affected by the output filter, supressing DC and higher harmonics of ω_o from the output. The filter bandwidth is large compared with the multicarrier signal bandwidth, so that the effect to be considered here is only the group delay τ in the filter.

The RF output signal of the transmitter is then found to be

$$v_1(a,t) = a\, g(a) e^{j\phi(a)}\, e^{-j\omega\tau} \cos \theta(t) \tag{7}$$

The intermodulation may be found in the following way: the ideal output would be

$$v_{1u}(a,t) = g_o a(t) \cos(\theta(t)) e^{j\phi_o}\, e^{-j\omega\tau} \tag{8}$$

where G_o and ϕ_o are constant gain and phase values, averages of G(a) and $\phi(a)$. The distortion is then

$$\Delta v_1(a,t) = a(t)\,(g(a)e^{j\phi(a)} - g_o e^{j\phi_o})\,e^{-j\omega\tau}\cos\theta(t)$$

$$= (\frac{g(a)}{g_o}\,e^{j(\phi(a)-\phi_o)} - 1)\,v_{1u}(a,t) \qquad (9)$$

This shows clearly that distortion is both phase and gain dependent.

Elimination of intermodulation noise is an exercise in reducing ΔV_1 to zero over the dynamic range of the input signal. This is generally not possible with a high number of carriers and efficiently operated devices. Linearisation by predistortion (Ref. 3) is complicated to realise at microwave frequencies, and for multistage transistor amplifiers different sets of stages have too much spread in their characteristics for a single compensator to be applied.

In contrast to this, feed-back techniques are inherently insensitive to parameter variations and could then be the basis of efficient intermodulation suppressing systems.

Below two feed-back schemes for improvement of the carrier to intermodulation ratio (C/I) are examined.

GAIN CONTROL WITH SHAPING

The principle is, as shown in Fig. 2, to compare samples of the input envelope with a fraction of the output envelope. Via a loop filter the difference is used to control a modulator in front of the power amplifier. In order to control the saturation of the amplifier the input sample is shaped. The feed-back loop may be characterised as being nonlinear, and in addition it also contains a significant delay. Both of these features influence the detail loop design, which is described extensively in an earlier paper (Ref. 4).

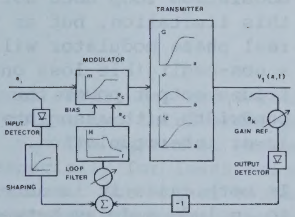


Figure 2. Power amplifier with envelope feedback loop.

Nonlinearity in the loop is not only determined by the amplifier characteristics but also by the characteristics of modulator and detectors. This means that when choosing these from available hardware one may try to reduce the total nonlinearity. Other factors to consider are operating points for modulator and detectors which may influence the dynamic range over which the control is active.

The main parameter determining the loop filter is the delay in the amplifier, which effectively sets an upper limit for the loop bandwidth obtainable.

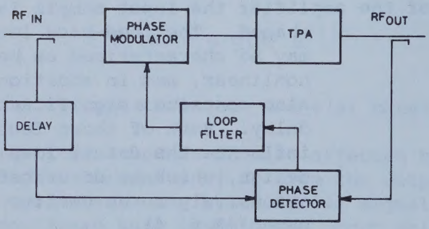
A shaped gain control (SGC) loop has been built and optimized for a L-band transistor power amplifier delivering 67 W. The study showed the following points to be of importance for good loop operation:

- Amplifier saturation must be avoided by correct setting of shaping, which may be a pure limiter.
- Dynamic range of loop must be as large as possible. This is controlled mainly by nominal modulator gain and reference gain setting.
- Loop bandwidth and DC-gain should be as large as possible.

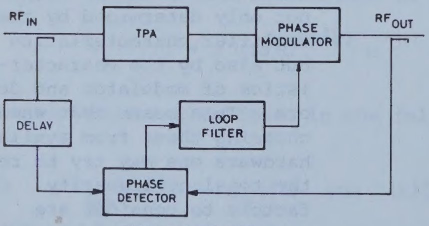
The optimum was found to be "flat" so that the results shown in the section SIMULATIONS AND RESULTS are representative over a range of parameter variations.

PHASE CONTROL

The intermodulation generated by the phase being a function of input power can to some extent be suppressed by a phase controller with a layout as one of the two shown in Fig. 3. As an ideal phase modulator is without loss, (it could for instance be a variable capacitive load), the output modulator scheme is a possibility which may be considered. This scheme has one great advantage as may be seen when the characteristics of the two loops are compared. In both cases the loop components are linear. But in the input modulation case the feed-back loop contains the delay of the amplifier. This is setting the limit for the loop bandwidth. The output modulation loop does not have this limitation, but as the real phase modulator will have a non-negligible loss one trades output power for large bandwidth with associated lower intermodulation.



INPUT MODULATION



OUTPUT MODULATION

FIG.3 ALTERNATIVE PHASE CONTROL LOOPS

In both cases it is necessary to include a delay between RF input and the phase detector. The delay must be matched to the amplifier delay in order to avoid input phase variation being converted into phase distortion. This conversion is determined also by loop gain so that with a better delay-match the larger loop band-width can be tolerated.

The shaped gain controller and the phase controller may be implemented to work on the same time as they are controlling independent properties For ideal loops, i.e. no loop errors, this combined system realises a power amplifier with the shaping function as its characteristic. This function thus determines the limit for achievable intermodulation suppression.

Results from simulation are given in the following section for both loops and the combination.

SIMULATION AND RESULTS

The two types of loops for intermodulation reduction have been examined in detail by analog computer simulations. The heart of the simulation is the power amplifier model shown in Fig. 4. The model is a realization of equation (7) made by detection of a(t) and Θ(t) (1) of the input signal and using a(t) to determine output amplitude and phase distortion. Essential for the set up of the simulation is the fact that the actual value of the centre frequency is unimportant for all intermodulation characteristics. This means that a very low value (2.2 KHz) of ω_0 compared to the frequency spacing between carriers (50 Hz) may be used in order to facilitate computer scaling and better utilization of analog computer components.

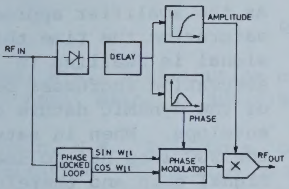


FIG.4 BASIC MODEL OF POWER AMPLIFIER

The validity of the amplifier model and the simulation was checked by comparing simulated results with results from hardware measurements. The device used for comparison was a transistor amplifier of make FIAR with a saturated output power of 12.5 W. The input signal was three carriers with equal amplitudes and a ratio between frequency spacings of 1 to 6. In the output two third order products were measured with the resul shown in Fig. 5. The agreement between hardware and simulation is seen to be good, especially for lower values of the carrier to intermodulation power (C/I) ratios. For higher values of C/I, the computer and hardware results diverge probably because computer accuracy is becoming too poor.

For the loop measurement an input signal composed of 12 equally spaced, equal amplitude carriers with independent phase modulation was used. In order to measure intermodulation power, one of the carriers in the centre of the band was removed. This complicated input signal was generated off-line and recorded on tape-recorder for later play-back into the simulation set-ups.

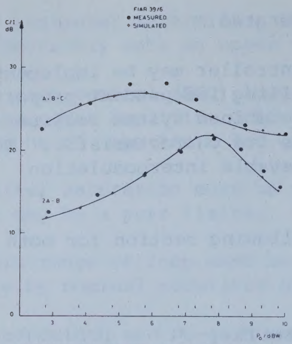


FIG 5 COMPARATIVE SIMULATION AND HARDWARE RESULTS

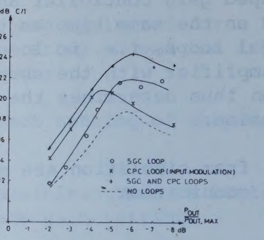


FIG 6 PERFORMANCE IMPROVEMENTS USING PCL AND SGC

Figures 6 and 7 give the simulation results obtained by closing the loops around an amplifier with characteristics as shown in Fig. 1. These characteristics are those of a 67 Watt solid state amplifier operating at C-Band built for the MAROTS satellite by MARCONI. The values of C/I are plotted against a decrease in output power from saturation, P_{max}.

As can be judged from the effect of the PCL loop when closed, the phase distortion is dominated by the amplitude distortion at high back-off and is slightly dominant near saturation.

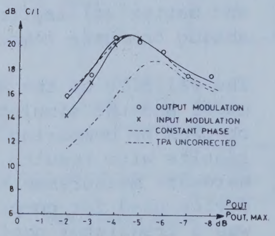


FIG 7 PERFORMANCE IMPROVEMENT USING PHASE CONTROL (PCL)

As the amplifier approaches saturation the time that the signal is dwelling in saturation increases because of the dynamic nature of the envelope. When in saturation the SGC loop has no small signal gain and therefore provides no benefit. This it the cause of the decreasing loop performance as can be been in Fig. 6.

In Fig. 6 the phase control (PCL) with input modulation was used. A comparison with output modulation is given in Fig. 7, again as C/I versus output power. For high output powers the difference is seen to be large, with the output modulation scheme being of the same quality as if the phase was in fact constant and no loop applied.

CONCLUSION

The paper has described application of feed-back to reduce intermodulation generated in nonlinear microwave amplifiers under multicarrier operation. As an example a transistor power amplifier was studied by simulation. Significant improvements were found as the result of using one of the two feed-back schemes or by using both. The characteristics of the schemes were found to be different in respect to the power level at which the loops were most efficient, the phase controller being best near saturation and the gain controller away from saturation.

These conclusions are based on transistor amplifier characteristics and may of course not hold unchanged for any type of power amplifier. But the authors believe that by knowing the relative merits of the two loop concepts and the characteristics of a given amplifier, it should not be too difficult to decide if one of the loops might help one design a system fulfilling given output power and intermodulation constraints.

ACKNOWLEDGEMENT

Grateful acknowledgements are made to the European Space Agency, ESTEC division, Holland with which the authors were associated during the work leading to the paper. In particular to the applied mathematics group and M. Guerin for his wise council which in a large way contributed to the completion of the paper.

REFERENCES

1. Bennet, W.R., Distribution of the sum of randomly phased components, Q. of Appl. Math. 5, 385 (1948).

2. Blachman, N.M., Detectors, band pass nonlinearities and their optimisation - Inversion of the Chebychev transform. IEEE Trans. Inf. Theory IT-17, 398 (1971).

3. Kaye, A.R. et al., Analyses and compensation of bandpass nonlinearities for communications. IEEE Trans. Commun. COM-20, 965 (1972).

4. Blanke, M., McElhone, T.P., Hybrid analog computer techniques; their use in designing and assessing the performance of nonlinear transmitters under multicarrier operation. ESA publication STM-205 (1978).

LOW-NOISE REALIZATIONS FOR DIGITAL FILTERS WITH POLES NEAR THE UNIT CIRCLE*

DAVID C. MUNSON, JR. AND BEDE LIU
Dept. of Electrical Engineering and Computer Science
Princeton University, Princeton, New Jersey 08540

ABSTRACT

Agarwal and Burrus have proposed two all-pole digital filter structures which have very small roundoff error for poles near the unit circle. ROM-accumulator (ROM/ACC) implementations of these structures have also been shown to have small error. We extend these results by including zeroes in both the Agarwal-Burrus structures and their ROM/ACC counterparts. A modified approach toward scaling and selecting various wordlengths is required. Expressions for MSE are derived for the new structures and shown to be small.

I. INTRODUCTION

Arithmetic roundoff error and coefficient quantization can be very large for direct form realizations of digital filters with poles near the unit circle [1-5]. However, in two all-pole filter structures proposed by Agarwal and Burrus [3] these errors were shown to be considerably reduced. Since multiplierless ROM-accumulator (ROM/ACC) implementations of digital filters are often efficient in terms of speed and hardware [6-8], it was of interest to study ROM/ACC implementations of the Agarwal-Burrus structures. This was subsequently done by Munson and Liu in [4,5] in which the new ROM/ACC structures were shown to possess the same improved error properties demonstrated by Agarwal and Burrus.

In this paper we extend the above results by including zeroes in both conventional and ROM/ACC implementations. To this end consider the narrowband filter with transfer function

$$G(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{z^2 + b_1 z + b_2} \qquad (1)$$

with poles at $z = (1-\delta)e^{\pm j\theta}$ so that

$$b_1 = -2(1-\delta)\cos\theta \ , \ b_2 = (1-\delta)^2 \ . \qquad (2)$$

Following Agarwal and Burrus we note that for poles close to the unit circle $b_2 \approx 1$, so it is advantageous to realize b_2 as $(1-\tilde{b}_2)$ where $\tilde{b}_2$ is a small positive quantity. If in addition the poles are close to z=1 (narrowband lowpass), then $b_1 \approx -2$ and it can be realized as $(-2 + \tilde{b}_1)$ where $\tilde{b}_1$ is small and positive. For such a realization

---

*This research was sponsored by the Air Force Office of Scientific Research, USAF under Grant AF-AFOSR-76-3083 and by the National Science Foundation under Grant ENG-76-19808.

$$\tilde{b}_1 = 2 + b_1 \; , \; \tilde{b}_2 = 1 - b_2 \; . \tag{3}$$

For poles near z=1 the pole section could alternatively be realized by making the change of coordinates [3] $\hat{z} = z-1$ in the denominator of (1). The transfer function then becomes

$$G(z) = \hat{z}^{-2} \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + \hat{b}_1 \hat{z}^{-1} + \hat{b}_2 \hat{z}^{-2}} \tag{4}$$

where

$$\hat{b}_1 = 2 + b_1 = \tilde{b}_1 \; , \; \hat{b}_2 = 1 + b_1 + b_2 \tag{5}$$

and $\hat{b}_1$ and $\hat{b}_2$ are small. The quantity $\hat{z}^{-1}$ can be implemented by an accumulator followed with a delay.

From [3-5] one expects that an implementation of (1) making use of either of the above approaches will have much lower error than a straightforward direct form realization. However, an efficient scaling policy, including appropriate register lengths, is not immediately apparent, especially for the ROM/ACC realizations. This difficulty is resolved in that a scaling policy is suggested and wordlengths are chosen so that the total effect of finite wordlength error at the filter output is reduced. Fortunately the suggested scaling policy also results in expressions for MSE that are easily compared for narrowband lowpass filters in general.

In Secion II we compare a direct form second order filter section using conventional arithmetic with conventional arithmetic realizations of the Agarwal-Burrus structures modified to include zeroes. Assuming poles near z=1, expressions for mean squared error are given. Mean squared errors for the new pole-zero realizations are shown to be much smaller than the error for the direct form realization. One of the new structures may be easily modified for the more general case with poles near the unit circle. Section III parallels Section II except all realizations use ROM/ACC arithmetic. The new ROM/ACC realizations are shown to have errors of comparable magnitude to their corresponding conventional arithmetic realizations and much lower error than a ROM/ACC direct form realization. As a consequence, the new pole-zero structures can offer significant hardware and speed advantages.

II. CONVENTIONAL ARITHMETIC REALIZATIONS AND MSE FOR POLE PAIR NEAR z=1

A. Realization DFZ[1]

Fig. 1 depicts a direct form realization of a second order filter section included for comparison purposes. The filter is scaled by G_p where for L^∞ scaling we take

$$G_p = \max_\lambda |G(e^{j\lambda})| \; . \tag{6}$$

---

[1] Direct form with zeroes.

For some filters L^{∞} scaling may be slightly optimistic, but the following results are all easily modified for alternate scaling. In Fig. 1 the error e_n^1 results from roundoff following the five multiplications. Assuming (B+1) bit rounding arithmetic it can be shown [9] that for small δ and θ the mean squared arithmetic roundoff error is approximately

$$N \approx \frac{5G_p^2}{48\delta(\theta^2 + \delta^2)} 2^{-2B} \tag{7}$$

which can be very large.

B. Realization 1ABZ

Realization 1ABZ[2] illustrated in Fig. 2 is a modification of Realization 1AB in [4,5] obtained by adding a zero section and altering the scaling. It is a good structure for the case with poles near z=1 and can also be used for the more general situation with poles near the unit circle if the multiplication by b_1 is performed directly [3].

The small error in Realization 1AB was a consequence of not only the basic filter structure, but was also due to the use of internal scaling (as opposed to scaling only at the filter input and output), which required additional wordlength for some registers in order to prevent overflow. Attempting to take advantage of similar features in Realization 1ABZ, we use internal scaling and also scale up the input by a factor L. This additional scaling by L provides an input to the pole section in Realization 1ABZ having magnitude more comparable to the input magnitude in Realization 1AB. Therefore, additional wordlength for some registers in the pole section of Realization 1ABZ will be required. Specifically, assuming (B+1) bit multipliers in Fig. 2, the adder and two feedback delay registers require $(B+1+\lceil \log_2 LG_p \rceil)$ bits. The error e_n^1 is due to roundoff following the five multiplications by La_0, La_1, La_2, $-LG_p\tilde{b}_1$, and $LG_p\tilde{b}_2$. The errors e_n^2 and e_n^3 result from scaling by $1/LG_p$. It is shown in [9] that a good choice for L is

$$L = \left[\frac{G_p^2}{5} (\theta^4 + 4\delta(\theta^2 + \delta^2)) \right]^{-\frac{1}{2}} \tag{8}$$

which equates the mean squared error at the filter output due to e_n^1 with the sum of the MSE's due to the errors e_n^2 and e_n^3. A larger L will require longer wordlengths for the adder and two feedback delay registers and will not substantially reduce the total error at the output.

It is also shown in [9] that for small δ and θ the mean squared error for Realization 1ABZ is approximately

---

[2] First Agarwal-Burrus structure with zeroes.

$$N \approx \frac{G_p^2}{6}\left[1 + \frac{\theta^4}{4\delta(\theta^2 + \delta^2)}\right] 2^{-2B} \quad . \tag{9}$$

This is obviously much smaller than the corresponding expression for Realization DFZ given by (7).

C. Realization 2ABZ

Realization 2ABZ[3] in Fig. 3 is advantageous for the case with poles near z=1. This realization is a modification of Realization 2AB in [4,5] obtained by adding a zero section and altering the scaling. The error source e_n^1 is due to roundoff following the five non-scaling multiplications. Errors e_n^2 and e_n^3 result from scaling by $1/LG_{1p}$ and G_{1p}/G_p respectively, where

$$G_{1p} = \max_{\lambda} |G_1(e^{j\lambda})| \tag{10}$$

and

$$G_1(z) = (z-1)G(z) \tag{11}$$

is 1/L times the transfer function from the input to s_n. The scaling value L is again chosen so that the mean squared error at the output due to e_n^1 is approximately equal to the sum of the MSE's due to e_n^2 and e_n^3. From [9] the value chosen for L is

$$L = \left[\frac{G_{1p}^2}{5}(\theta^2 + \delta^2) + \frac{G_p^2}{5}(\theta^4 + 4\delta(\theta^2 + \delta^2))\right]^{-\frac{1}{2}} \tag{12}$$

Inputs to the multipliers $(-LG_{1p}\hat{b}_1)$ and $(-LG_p\hat{b}_2)$ are scaled to (B+1) bits. However, the adder and second $\hat{z}^{-1}$ block require $(B+1+\lceil \log_2 LG_{1p}\rceil)$ and $(B+1+\lceil \log_2 G_p/G_{1p}\rceil)$ bits respectively where it is assumed that the adder performs the addition required by the first $\hat{z}^{-1}$ block. It can then be shown [9] that for small δ and θ the mean squared error for Realization 2ABZ is approximately

$$N \approx \frac{1}{6}\left[\frac{G_{1p}^2}{4\delta} + G_p^2\left(1 + \frac{\theta^4}{4\delta(\theta^2 + \delta^2)}\right)\right] 2^{-2B} \quad . \tag{13}$$

This is seen to be of the same order of magnitude as the MSE expression for Realization 1ABZ given by (9) and is considerably smaller than the expression for Realization DFZ given by (7).

III. ROM/ACC REALIZATIONS AND MSE FOR POLE PAIR NEAR z=1

A. ROM/ACC Implementation of Realization DFZ

The implementation of Realization DFZ with ROM/ACC arithmetic is shown as Realization DFZR[4] in Fig. 4. Neglecting error, the ROM/ACC arithmetic unit consisting of a ROM and an accumulator calculates

---

[3] Second Agarwal-Burrus structure with zeroes.

[4] Direct form with zeroes using ROM/ACC.

$$\hat{r}_n = \frac{1}{G_p}(a_0x_n + a_1x_{n-1} + a_2x_{n-2}) - b_1v_n - b_2w_n \ . \tag{14}$$

As in [6,7] it performs this computation in a bitwise manner, namely

$$\hat{r}_n = -\varphi(x_n^0, x_{n-1}^0, x_{n-2}^0, v_n^0, w_n^0) + \sum_{j=1}^{B} 2^{-j}\varphi(x_n^j, x_{n-1}^j, x_{n-2}^j, v_n^j, w_n^j) \tag{15}$$

where the binary value x_n^j is equal to the jth bit in the two's complement (B+1) bit representation of x_n and

$$\varphi(x_n^j, x_{n-1}^j, x_{n-2}^j, v_n^j, w_n^j) = \frac{1}{G_p}(a_0x_n^j + a_1x_{n-1}^j + a_2x_{n-2}^j) - b_1v_n^j - b_2w_n^j . \tag{16}$$

The 32 possible values of $\varphi(\cdot)$ are scaled down by

$$W = \max|\varphi(x_n^j, x_{n-1}^j, x_{n-2}^j, v_n^j, w_n^j)| \tag{17}$$

for storage in finite wordlength ROM. Taking W=2 will generally be sufficient, although elliptic filters will often require W=4. We further assume (B+1+C) bit ROM where C is chosen to balance the effects of the errors e_n^1 and e_n^2 in Fig. 4. The error e_n^1 is due to roundoff of the sum in (15) to (B+1) bits and e_n^2 results from storage of the values of $\varphi(\cdot)$ in finite wordlength ROM. From [9] an approximate value for C is

$$C \approx \log_2 \frac{\sqrt{5}}{2} W \ . \tag{18}$$

Of course in an actual realization C would be an integer, and in fact for Realization DFZR it would generally be taken as zero. However, it is included here for uniformity since in some cases the ROM/ACC realizations to be described shortly will require several additional bits for ROM wordlength. It is further assumed that (B+1+max{0,C}) bits are used for the accumulator wordlength to assure B significant fractional bits in the ROM/ACC unit.

Using the above value for C, it can be shown [9] that for small δ and θ the mean squared error for Realization DFZR is approximately

$$N \approx \frac{G_p^2}{24\delta(\theta^2 + \delta^2)} 2^{-2B} \ . \tag{19}$$

This expression can be very large and is nearly identical to its counterpart (7) for Realization DFZ.

B. ROM/ACC Implementation of Realization 1ABZ

The multiplications in Fig. 2 by La_0, La_1, La_2, $-LG_p\tilde{b}_1$, and $LG_p\tilde{b}_2$ can be implemented with a ROM/ACC arithmetic unit resulting in the structure labelled Realization 1ABZR[5] in Fig. 5. The errors e_n^1 and e_n^4 have the same interpretations as e_n^1 and e_n^2 respectively in the previous Realization DFZR. The errors e_n^2 and e_n^3 result from scaling by $1/LG_p$. Again we assume (B+1+C) bit ROM wordlength. The values L and C are chosen so that the errors at the output due to e_n^1 and e_n^4 are each equal to the sum of the errors due to e_n^2 and e_n^3. From [9]

---

[5] First Agarwal-Burrus structure with zeroes using ROM/ACC.

$$L = \left[G_p^2(\theta^4 + 4\delta(\theta^2 + \delta^2)) \right]^{-\frac{1}{2}} \tag{20}$$

and C is again given by (18) where now

$$W = L \cdot \max | a_0 x_n^j + a_1 x_{n-1}^j + a_2 x_{n-2}^j - G_p \tilde{b}_1 v_n^j + G_p \tilde{b}_2 w_n^j | . \tag{21}$$

The accumulator and the delay registers in the pole section will require $(B+1+\max\{C, \lceil \log_2 LG_p \rceil\})$ and $(B+1+\lceil \log_2 LG_p \rceil)$ bits respectively. Assuming the above, it can be shown [9] that for small δ and θ the mean squared error for Realization 1ABZR is approximately

$$N \approx \frac{G_p^2}{4} \left[1 + \frac{\theta^4}{4\delta(\theta^2 + \delta^2)} \right] 2^{-2B} . \tag{22}$$

This expression is comparable to the same quantity (9) for Realization 1ABZ and can be much smaller than the corresponding error for Realization DFZR given by (19).

Filters with zeroes at z= -1 will have small $\{a_i\}$, so from (18) and (21) it can be seen that C will be relatively small for Butterworth, Chebychev, and Bessel Filters. However, the zeroes of a narrowband elliptic filter are close to the poles and hence O(1). The corresponding C may be several bits for this type of filter.

C. ROM/ACC Implementation of Realization 2ABZ

The non-scaling multiplications in Fig. 3 can be implemented with a ROM/ACC arithmetic unit resulting in the structure labelled Realization 2ABZR[6] in Fig. 6. Each error source e_n^i has an interpretation similar to its previously described counterpart in Fig. 5. As before, C additional bits are used for ROM wordlength. From [9] the various errors will be balanced at the output if

$$L = \left[G_{1p}^2(\theta^2 + \delta^2) + G_p^2(\theta^4 + 4\delta(\theta^2 + \delta^2)) \right]^{-\frac{1}{2}} \tag{23}$$

and again C is taken as given by (18) where

$$W = L \cdot \max | a_0 x_n^j + a_1 x_{n-1}^j + a_2 x_{n-2}^j - G_{1p} \hat{b}_1 v_n^j - G_p \hat{b}_2 w_n^j | . \tag{24}$$

The accumulator and second $\hat{z}^{-1}$ block will require $(B+1+\max\{C, \lceil \log_2 LG_{1p} \rceil\})$ and $(B+1+\lceil \log_2 G_p/G_{1p} \rceil)$ bits respectively. As before, it can be seen that C will be larger for an elliptic filter than for filters with zeroes at z= -1.

---

[6] Second Agarwal-Burrus structure with zeroes using ROM/ACC.

Assuming the above it can be shown [9] that for small δ and θ the mean squared error for Realization 2ABZR is approximately

$$N \approx \frac{1}{4}\left[\frac{G_{1p}^2}{4\delta} + G_p^2\left(1 + \frac{\theta^4}{4\delta(\theta^2 + \delta^2)}\right)\right]2^{-2B} . \qquad (25)$$

This expression is comparable to the same quantity (13) for Realization 2ABZ and is much smaller than the corresponding error for Realization DFZR given by (19).

Table I compares the mean squared error expressions for all realizations. These expressions simplify somewhat for the frequent case with $\theta^2 \gg \delta^2$, facilitating comparison. The MSE for this case is listed in the second column where it is also assumed that $G_{1p} \approx G_p\theta$. This assumption holds for many cases of interest as explained in [9].

IV. DISCUSSION

We have considered the realization of narrowband (primarily lowpass) second order digital filter sections. The large roundoff error inherent in direct form realizations of such filters was previously shown to be considerably reduced in two all-pole structures proposed by Agarwal and Burrus [3]. ROM/ACC implementations of these structures were later analyzed in [4,5]. Here we extended these results by including zeroes in both the Agarwal-Burrus structures and their ROM/ACC counterparts. It was necessary, however, to take an approach somewhat different from [4,5] toward scaling and selecting register lengths. The fundamental differences were that first, the input was scaled up by a factor L to reduce the effect of e_n^1. Secondly, C additional bits were used for ROM wordlength, since unlike the all-pole case it was found that the error due to finite wordlength ROM could be clearly dominant. This was especially true for elliptic filters. We parenthetically remark that using additional ROM wordlength does not affect the word rate of the filter, since it is inversely proportional to the wordlength at the ROM/ACC unit input, not the ROM wordlength itself.

Assuming the modified scaling and wordlength requirements, expressions for MSE were derived for the new pole-zero realizations and shown to be small compared with corresponding expressions for either direct form conventional arithmetic or ROM/ACC filters. As a consequence the new realizations can offer substantial hardware and speed advantages.

REFERENCES

[1] B. Liu, "Effect of finite word length on the accuracy of digital filters-A review," IEEE Trans. Circuit Theory, Vol. CT-18, pp.670-677, Nov. 1971.

[2] A.V. Oppenheim and C.J. Weinstein, "Effects of finite register length in digital filtering and the fast Fourier transform," Proc. IEEE, Vol. 60, pp.957-976, Aug. 1972.

[3] R.C. Agarwal and C.S. Burrus, "New recursive digital filter structures having very low sensitivity and roundoff noise," IEEE Trans. Circuits Syst., Vol. CAS-22, pp.921-927, Dec.1975.

[4] D.C. Munson, Jr. and B. Liu, "ROM realization of digital filters for poles near the unit circle," in Proc. 1978 International Symp. on Circuits and Systems, New York, NY, May 17-19, 1978.
[5] D.C. Munson, Jr. and B. Liu, "ROM/ACC realization of digital filters for poles near the unit circle," submitted to IEEE Trans. Circuits Syst.
[6] A. Peled and B. Liu, "A new hardware realization of digital filters," IEEE Trans. Acoust., Speech, and Signal Processing, Vol. ASSP-22, pp.456-462, Dec. 1974.
[7] A. Croisier, D.J. Esteban, M.E. Levilion, and V. Riso, "Digital filter for PCM encoded signals," U.S. Patent 3 777 130, Dec. 3, 1973.
[8] S.A. White, "Applications of digital signal processing to control systems," in Proc. Eighth Asilomar Conf. on Circuits, Systems, and Computers, Pacific Grove, CA, pp.278-284, Dec. 3-5, 1974.
[9] D.C. Munson, Jr. and B. Liu, "Low-noise realizations for narrowband recursive digital filters," to be submitted.

TABLE I

MSE FOR VARIOUS REALIZATIONS FOR SMALL δ and θ

Realization	MSE	MSE for $\theta^2 >> \delta^2$
DFZ	$\frac{5G_p^2}{48\delta(\theta^2+\delta^2)} 2^{-2B}$	$\frac{5G_p^2}{48\delta\theta^2} 2^{-2B}$
1ABZ	$\frac{G_p^2}{6}[1 + \frac{\theta^4}{4\delta(\theta^2+\delta^2)}] 2^{-2B}$	$\frac{G_p^2}{6}[1 + \frac{\theta^2}{4\delta}] 2^{-2B}$
2ABZ	$\frac{1}{6}[\frac{G_{1p}^2}{4\delta} + G_p^2(1 + \frac{\theta^4}{4\delta(\theta^2+\delta^2)})] 2^{-2B}$	* $\frac{G_p^2}{6}[1 + \frac{\theta^2}{2\delta}] 2^{-2B}$
DFZR	$\frac{G_p^2}{24\delta(\theta^2+\delta^2)} 2^{-2B}$	$\frac{G_p^2}{24\delta\theta^2} 2^{-2B}$
1ABZR	$\frac{G_p^2}{4}[1 + \frac{\theta^4}{4\delta(\theta^2+\delta^2)}] 2^{-2B}$	$\frac{G_p^2}{4}[1 + \frac{\theta^2}{4\delta}] 2^{-2B}$
2ABZR	$\frac{1}{4}[\frac{G_{1p}^2}{4\delta} + G_p^2(1 + \frac{\theta^4}{4\delta(\theta^2+\delta^2)})] 2^{-2B}$	* $\frac{G_p^2}{4}[1 + \frac{\theta^2}{2\delta}] 2^{-2B}$

(*) Also assumes $G_{1p} \approx G_p\theta$.

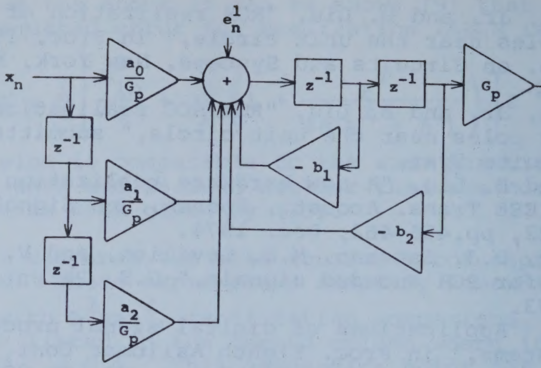

e_n^1
x_n
$\frac{a_0}{G_p}$
+
z^{-1}
z^{-1}
G_p
y_n
z^{-1}
$- b_1$
$\frac{a_1}{G_p}$
$- b_2$
z^{-1}
$\frac{a_2}{G_p}$

Fig. 1 Realization DFZ

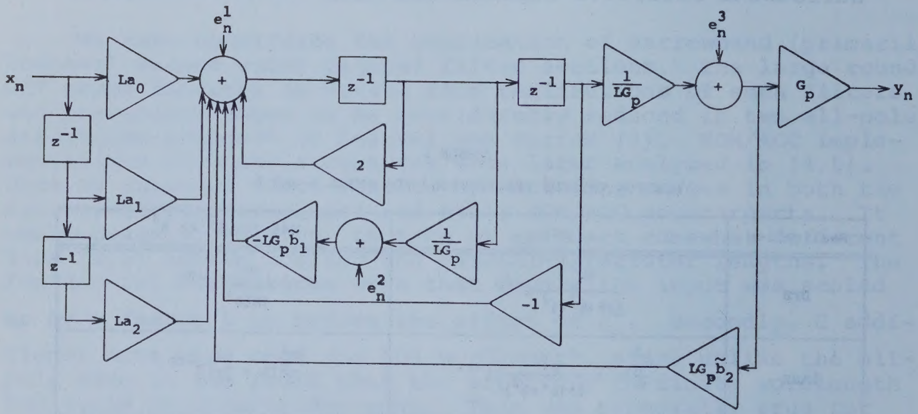

e_n^1
e_n^3
x_n
La_0
+
z^{-1}
z^{-1}
$\frac{1}{LG_p}$
+
G_p
y_n
z^{-1}
2
La_1
z^{-1}
$-LG_p\tilde{b}_1$
+
$\frac{1}{LG_p}$
e_n^2
-1
La_2
$LG_p\tilde{b}_2$

Fig. 2 Realization 1ABZ

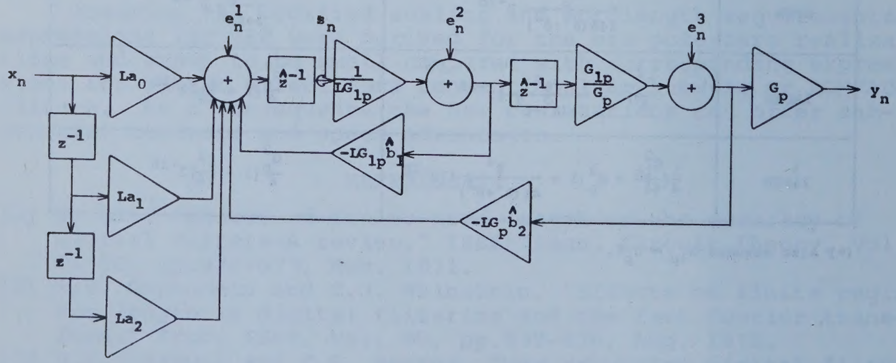
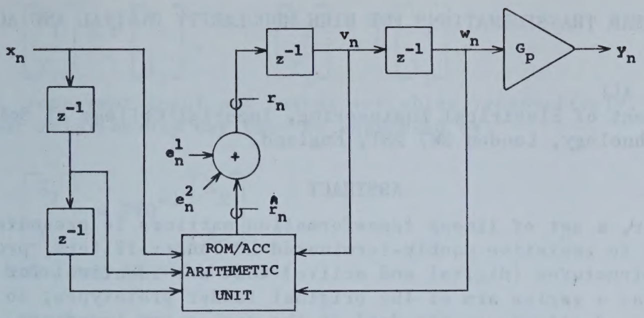
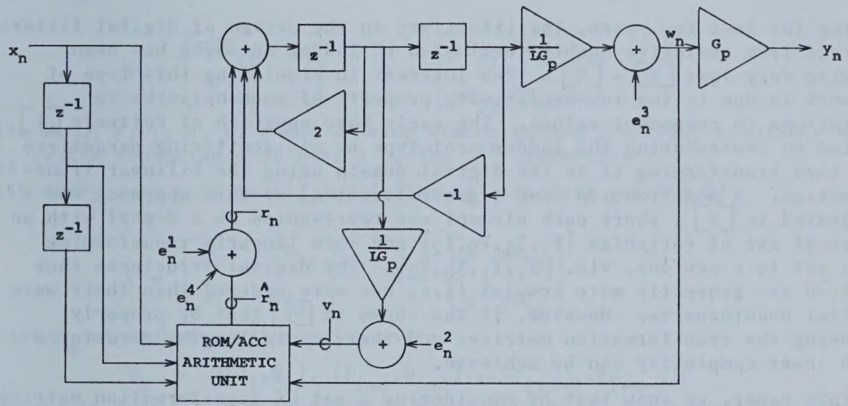

e_n^1
s_n
e_n^2
e_n^3
x_n
La_0
+
$\hat{z}^{-1}$
$\frac{1}{LG_{1p}}$
+
$\hat{z}^{-1}$
$\frac{G_{1p}}{G_p}$
+
G_p
y_n
z^{-1}
$-LG_{1p}\hat{b}_1$
La_1
$-LG_p\hat{b}_2$
z^{-1}
La_2

Fig. 3 Realization 2ABZ


x_n
z^{-1}
z^{-1}
z^{-1}
v_n
z^{-1}
w_n
G_p
y_n
r_n
e_n^1
+
e_n^2
$\hat{r}_n$
ROM/ACC
ARITHMETIC
UNIT

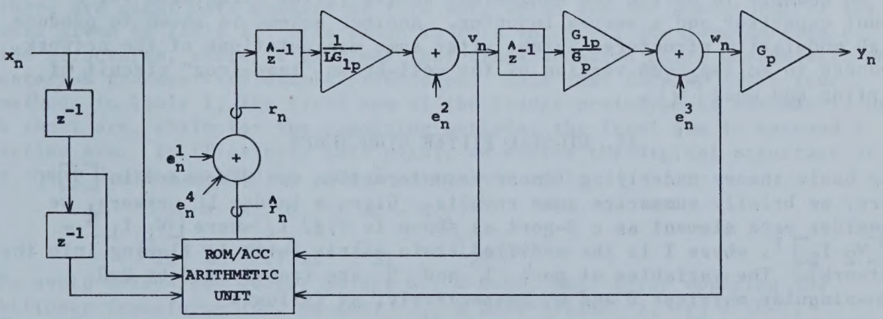
$$\hat{r}_n = \frac{1}{G_p}(a_0x_n + a_1x_{n-1} + a_2x_{n-2}) - b_1v_n - b_2w_n$$

Fig. 4 Realization DFZR


x_n
z^{-1}
z^{-1}
+
z^{-1}
z^{-1}
$\frac{1}{LG_p}$
+
w_n
G_p
y_n
e_n^3
2
-1
r_n
e_n^1
+
e_n^4
$\hat{r}_n$
$\frac{1}{LG_p}$
v_n
ROM/ACC
ARITHMETIC
UNIT
+
e_n^2

$$\hat{r}_n = L(a_0x_n + a_1x_{n-1} + a_2x_{n-2} - G_p\tilde{b}_1v_n + G_p\tilde{b}_2w_n)$$

Fig. 5 Realization 1ABZR


x_n
z^{-1}
z^{-1}
$\hat{z}^{-1}$
$\frac{1}{LG_{1p}}$
+
v_n
e_n^2
$\hat{z}^{-1}$
$\frac{G_{1p}}{G_p}$
+
w_n
e_n^3
G_p
y_n
r_n
e_n^1
+
e_n^4
$\hat{r}_n$
ROM/ACC
ARITHMETIC
UNIT

$$\hat{r}_n = L(a_0x_n + a_1x_{n-1} + a_2x_{n-2} - G_{1p}\hat{b}_1v_n - G_p\hat{b}_2w_n)$$

Fig. 6 Realization 2ABZR

A SET OF LINEAR TRANSFORMATIONS FOR HIGH MODULARITY DIGITAL AND ACTIVE FILTERS

AMAR M. ALI
Department of Electrical Engineering, Imperial College of Science and Technology, London SW7 2BT, England.

ABSTRACT

In this paper, a set of linear transformation matrices is presented which, when applied to resistive doubly-terminated LC ladder filters, produces equivalent structures (digital and active) that are identical for a shunt arm as well as a series arm of the original ladder prototypes, so long as the shunt arm admittance is the dual of the series arm impedance. This is desirable for LSI fabrication, since the resulting filter will consist of a cascade of one basic building block. Some examples are worked out to illustrate the excellent low sensitivity of such structures.

I. INTRODUCTION

During the last few years, the literature on the design of digital filters derived from resistive doubly-terminated LC ladder networks has been growing very fast [1] - [5]. The interest in simulating this type of network is due to the low-sensitivity property of such networks to variations in component values. The early wave approach of Fettweis [1] relied on representing the ladder prototype by its scattering parameters and then transforming it to the digital domain using the bilinear transformation. A modification (and a generalization) of this approach was suggested in [4], where each element was represented as a 2-port with an external set of variables $\{V_1, I_1, V_2, I_2\}$ and then linearly transforming this set to a new one, viz. $\{X_1, Y_1, X_2, Y_2\}$. The digital structures thus derived are generally more complex (i.e. use more adders) than their wave digital counterparts. However, it was shown in [5] that by properly choosing the transformation matrices and their cycle length, structures with lower complexity can be achieved.

In this paper, we show that by considering a set of transformation matrices that is acyclic, digital structures with high modularity may be derived, suitable for LSI fabrication. In the next section, sixteen different schemes possessing this property will be identified and those producing structures with minimum complexity will be pointed out. The straightforward extension of the above procedure to the design of high modularity active filters is presented in section III, where one scheme is employed as an example to derive the equivalent active filter structures for a shunt capacitor and a series inductor. Another scheme is shown to produce high modularity structures which, after some manipulations of the network, reduces to an improved version of the well-known "leap-frog" circuit of Girling and Good [6].

II. DIGITAL FILTER STRUCTURES

The basic theory underlying linear transformation was discussed in [5]; here, we briefly summarize some results. Given a ladder LC network, we consider each element as a 2-port as shown in Fig. 1, where $[V_1 \; I_1]^T = T \, [V_2 \; I_2]^T$, where T is the modified chain matrix (with I_2 flowing into the network). The variables at port 1 and 2 are transformed by 2x2 non-singular matrices P and Q, respectively, as follows:

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = P \begin{bmatrix} V_1 \\ I_1 \end{bmatrix}, \quad \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = Q \begin{bmatrix} V_2 \\ I_2 \end{bmatrix} \tag{1}$$

where X_i Y_i represent input and output variables, respectively. The input-output relationship may then be expressed as:

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = PTQ^{-1} \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} \tag{2}$$

which, in turn, may be written as:

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \underline{\sigma} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{3}$$

expressing each of the outputs Y_1, Y_2 as a linear combination of the inputs X_1 and X_2. The general form of P and Q is given by [5]:

$$P = \begin{bmatrix} \alpha G_{1i} & \beta \\ \gamma G_{1i} & \lambda \end{bmatrix}, \quad Q = \begin{bmatrix} aG_{2i} & b \\ cG_{2i} & d \end{bmatrix} \tag{4}$$

with $\alpha, \beta, \gamma, \lambda$, a, b, c and d being dimensionless arbitrary constants and G_{ji} is the conductance of port j of the i^{th} element.
With every such pair in (4), we define an associate pair $(\tilde{P},\tilde{Q})$:

$$\tilde{P} = \begin{bmatrix} \beta & \alpha R_{1i} \\ \lambda & \gamma R_{1i} \end{bmatrix}, \quad \tilde{Q} = \begin{bmatrix} b & aR_{2i} \\ d & cR_{2i} \end{bmatrix} \tag{5}$$

A set S_m of transformation pairs, viz:

$$S_m = \left\{(P_i,Q_i),\ (P_{i+1},Q_{i+1})\ldots(P_{i+m-1},Q_{i+m-1})\right\}$$

is said to have a cycle length m if the matrix Q_{i+m-1} is compatible with P_i; otherwise S_m is said to be acyclic.
The cases m = 1 and 2 were considered in [5].

Now, we consider an acyclic set S_m which possesses the property of high modularity mentioned earlier. Eight members of this set are found in Table I, all of which lead to different realization schemes. In addition, there are eight more members of this set, which are the associates of the ones given in Table I. Also shown in this table are the σ-parameters of the structures, where it is clear that transformation pairs marked with an asterisk produce very simple structures. Note that for the first four methods in Table I, the first arm of the ladder prototype is assumed to be a shunt arm, while for the remaining methods, the first arm is assumed a series arm. To illustrate this point, we derive the digital structure of a shunt C using method 2; the σ-parameters are:

$$\sigma_{11} = R_{11}(G_{11} - sC - G_{21})$$
$$\sigma_{12} = -R_{11}G_{21},\ \sigma_{21} = \sigma_{22} = 1$$

To avoid delay-free loops, we set $G_{11} = C+G_{21}$ and, after applying the bilinear transformation, we get: $\sigma_{11} = 2\alpha Z^{-1}/(1+Z^{-1})$, $\sigma_{12} = \alpha-1$, $\sigma_{21} = \sigma_{22} = 1$, with $\alpha = R_{11}C$. The structure is given in Fig. 2. The structure for a series L is the same as the one above, but now α is given by: $\alpha = LG_{11}$ and the realizability constraint becomes $R_{11} = R_{21} + L$.

III. ACTIVE FILTER STRUCTURES

Although the transformation pairs listed in Table I are primarily chosen for the design of digital filter structures, they can be used to produce high modularity active filter structures. The same procedure as in the previous section is applicable here, except in this case there is no problem of delay-free loops and hence the realizability constraint does not apply. This gives us more flexibility in choosing the transformation pairs. As an illustration, consider method 6, where the σ-parameters are given by:

$$\sigma_{11} = (R_{21}+Z-R_{11})/D, \quad \sigma_{12} = -R_{11}/D, \quad \sigma_{21} = \sigma_{22} = R_{21}/D$$

with $D = R_{21} + Z$, and Z is a series impedance. Now, we choose $R_{21}=R_{11}=R$. (This is permissible since the problem of delay-free loops in digital structures does not arise in the case of active structures.) Therefore,

$$\sigma_{11} = Z/D, \quad \sigma_{12} = -R/D, \quad \sigma_{21} = \sigma_{22} = R/D, \quad D = R+Z.$$

The structure may be realized as follows: we first realize Y_2 as: $Y_2 = \frac{R}{R+Z}(X_1+X_2)$. From the above expressions for the σ-parameters, we may write $Y_1 = X_1-Y_2$. Hence, the equivalent active circuit is shown in Fig. 3, where the output Y_1' is half its actual value. However, this can be taken care of in a subsequent section by scaling Y_1' up by a factor of 2, as was done in Fig. 3 for obtaining Y_2. The value of resistor r is arbitrary, and $C = L/R^2$.

The above method allows the overall network to have equal-valued capacitors all of which are grounded, which is very desirable from the fabrication point of view. Although the above structure uses two operational amplifiers (per element), one of them is a buffer and hence is expected to have good performance, especially at high frequencies.

As a second example, let us consider a particularly interesting transformation set with cycle length of 2, given below:

$$P_1 = \begin{bmatrix} 0 & R \\ 1 & 0 \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 0 & R \\ -1 & 0 \end{bmatrix}$$
$$P_2 = \begin{bmatrix} -1 & 0 \\ 0 & -R \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & -R \end{bmatrix} \tag{6}$$

This set also produces structures with high modularity. Applying (6) to a fifth order low-pass ladder of Fig. 4, we arrive at the following σ-parameters for a shunt arm:

$$\sigma_{11} = \sigma_{12} = 1/RY, \quad \sigma_{21} = \sigma_{22} = -1/RY,$$

where $Y = sC$. This means that $Y_1 = -Y_2$. For a series arm, the structure is the same, except that Y is replaced by Z/R^2. The overall structure is given in Fig. 5a. In fact, this circuit may easily be reduced to the well-known "leap-frog" circuit [6] by proper displacement of the inverters (this may only cause a phase shift of 180° in the overall transfer function), shown in Fig. 5b. However, by replacing the first and last sections by a buffer and a grounded capacitor and then combining the inverters of the third and fourth arms into one appearing within the integrator path, we arrive at the circuit shown in Fig. 5c. (Note that a similar circuit to that of Fig. **5c** was arrived at in [9] but not in the context of high modularity transformations.) We suggest a further simplification of the latter circuit by replacing every integrator followed by an inverter with a positive integrator requiring only one OP AMP, one grounded capacitor and four resistors. The final circuit is shown in Fig. 5d, where only five OP AMPS are used, two of which act as buffers. In general, for an N^{th} order filter, the saving of OP AMPS is N/2, which is significant as OP AMPS not only contribute to the cost of the filter but also to power consumption and heat dissipation, which are required to be kept to a minimum for many practical applications.

IV. EXAMPLES

Example 1 Using the transformation pair of method 2, a fifth order, low-pass Elliptic filter C0520/47 [7] was designed. The passband response when the multiplier coefficients were truncated to 2 decimal points is shown in Fig. 6. Also shown is the response of the same filter designed using the approach of Sedlmeyer and Fettweis [8]. Clearly, in our case the ripple suffers no distortion and the bandwidth is unaltered, while in the wave digital filter case, there is an error of 8% in the ripple and the bandwidth suffered some contraction.

Example 2 A fifth order, low-pass Chebyshev filter was designed, as shown in Fig. 5d, and simulated (using a computer program) with OP AMPS modelled to have 1 MHz gain-bandwidth (GB) product. The cut-off frequency was 1 KHz and the passband response is shown in Fig. 7, which is in close agreement with the theoretical one. The leap-frog circuit of Fig. 5b was also simulated using the same type of OP AMPS and the resulting passband is shown in Fig. 7 also. It is clear that our circuit is less sensitive to the variations in the GB of the OP AMPS. Note that passive sensitivities in both cases are expected to be very low, since each capacitor in the active structure corresponds to only one reactive element of the ladder prototype.

CONCLUSIONS

We have presented a number of transformation matrices which produce different realization schemes for the design of digital filters with high modularity. Those schemes producing simple digital structures are identified. The procedure is then extended to include the realization of active filters with high modularity. One particular scheme led to an improved version of the well-known leap-frog circuit, using less number of operational amplifiers. The sensitivity of those structures is shown to be very low through two worked examples.

ACKNOWLEDGEMENTS

The author wishes to thank Professor Colin Cherry for his encouragement and support.

REFERENCES

[1] A. Fettweis, "Digital filter structures related to classical filter networks", Arch. Elek. Uebertragung, Vol. 25, No. 2, pp. 79-89, Feb. 1971.

[2] S.S. Lawson and A.G. Constantinides, "A method for deriving digital filter structures from classical filter networks", Proc. IEEE Symp. on Circuits and Systems, pp. 170-175, Sep. 1975

[3] L.T. Bruton and D.A. Vaughan-Pope, "Synthesis of digital ladder filters from LC filters", IEEE Trans. CAS-23, No. 6, pp. 395-402, June 1976.

[4] S.S. Lawson, "Digital filter structures from classical analog networks", Ph.D. Thesis, University of London, 1975.

[5] A.M. Ali and A.G. Constantinides, "Design of low sensitivity and complexity digital filter structures", 1978 European Conf. on Circuit Theory and Design, Lausanne, Switzerland, pp. 335-339, Sept. 1978.

[6] F.E.J. Girling and E.F. Good, "Active filters - a practical approach on feedback amplifier theory", Parts 12, 13 and 14. Wireless World, July/Sept. and October 1970.

[7] V.R. Saal, "Der Entwurf von Filtern mit Hilfe des Kataloges normierter Tiefpässe", Backrang/Württ: Telefunken GmbH, 1961.

[8] A. Sedlmeyer and A. Fettweis, "Realization of Digital Filters with true ladder configuration", Int. J. Circuit Theory and Applications, Vol. 1, No. 1, pp. 5-10, 1973.

[9] H.G. Dimopoulos and A.G. Constantinides, "The LTA Filters". To be published.

Method	i	P_i	Q_i	σ-parameters
1	odd	$\begin{matrix} 0 & -n \\ -nG_{1i} & -n \end{matrix}$	$\begin{matrix} -nG_{1i} & nG_i R_{2i} \\ -nG_{1i} & 0 \end{matrix}$	$\sigma_{11}=(G_{1i}+Y-G_{2i})/D$ $\sigma_{12}=-G_2/D$, $\sigma_{21}=G_{1i}/D$ $\sigma_{22}=-G_{2i}/D$, $D=Y-G_{2i}$
	even	$\begin{matrix} -n & 0 \\ -n & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & nR_{1i} \\ 0 & nR_{1i} \end{matrix}$	$\sigma_{11}=(Z+R_{1i}-R_{2i})/D$ $\sigma_{22}=-R_{2i}/D$, $\sigma_{21}=R_{1i}/D$ $\sigma_{22}=-R_{2i}/D$, $D=Z-R_{2i}$
2*	odd	$\begin{matrix} -nG_{1i} & 0 \\ -nG_{1i} & n \end{matrix}$	$\begin{matrix} nG_{1i} & nG_{1i}R_{2i} \\ 0 & nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11} = R_{1i}[G_{1i} - Y - G_{2i}]$ $\sigma_{12} = -R_{1i}G_{2i}$ $\sigma_{21} = \sigma_{22} = 1$
	even	$\begin{matrix} 0 & -nR_{1i} \\ n & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & -nR_{1i} \\ -nR_{1i}G_{2i} & 0 \end{matrix}$	$\sigma_{11} = G_{1i}[R_{1i} - R_{2i} - Z]$ $\sigma_{12} = -G_{1i}R_{2i}$ $\sigma_{21} = \sigma_{22} = 1$
3*	odd	$\begin{matrix} nG_{1i} & -n \\ 0 & -n \end{matrix}$	$\begin{matrix} -nG_{1i} & 0 \\ -nG_{1i} & -nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11} = \sigma_{12} = 1$ $\sigma_{21} = -G_{1i}R_{2i}$ $\sigma_{22} = R_{2i}[Y + G_{2i} - G_{1i}]$
	even	$\begin{matrix} -n & nR_{1i} \\ -n & 0 \end{matrix}$	$\begin{matrix} 0 & nR_{1i} \\ nR_{1i}G_{2i} & nR_{1i} \end{matrix}$	$\sigma_{11} = \sigma_{12} = 1$ $\sigma_{21} = -R_{1i}G_{2i}$ $\sigma_{22} = G_{2i}[R_{2i} + Z - R_{1i}]$
4	odd	$\begin{matrix} -nG_{1i} & -n \\ -nG_{1i} & 0 \end{matrix}$	$\begin{matrix} 0 & nG_{1i}R_{2i} \\ -nG_{1i} & nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11}=G_{1i}/D$, $\sigma_{12}=-G_{2i}/D$ $\sigma_{21}=G_{1i}/D$, $\sigma_{22}=(Y+G_{1i}-G_{2i})/D$ $D=G_{1i}+Y$
	even	$\begin{matrix} -n & -nR_{1i} \\ 0 & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & 0 \\ -nR_{1i}G_{2i} & nR_{1i} \end{matrix}$	$\sigma_{11}=R_{1i}/D$, $\sigma_{12}=-R_{2i}/D$ $\sigma_{21} = \sigma_{11}$, $\sigma_{22}=(Y+R_{1i}-R_{2i})/D$ $D = R_{1i}+Z$

TABLE 1

Method	i	P_i	Q_i	σ-parameters
5*	odd	$\begin{matrix} 0 & -n \\ -nG_{1i} & -n \end{matrix}$	$\begin{matrix} -nG_{1i} & nG_{1i}R_{2i} \\ -nG_{1i} & 0 \end{matrix}$	$\sigma_{11}=G_{1i}[R_{1i}-R_{2i}+Z]$ $\sigma_{12}=1,\ \sigma_{21}=-G_{1i}R_{2i}$ $\sigma_{22}=1.$
	even	$\begin{matrix} -n & 0 \\ -n & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & nR_{1i} \\ 0 & nR_{1i} \end{matrix}$	$\sigma_{11} = R_{1i}[G_{1i} + Y - G_{2i}]$ $\sigma_{12} = 1,\ \sigma_{21} = -R_{1i}G_{2i}$ $\sigma_{22} = 1$
6	odd	$\begin{matrix} -nG_{1i} & 0 \\ -nG_{1i} & n \end{matrix}$	$\begin{matrix} nG_{1i} & nG_{1i}R_{2i} \\ 0 & nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11} = (R_{2i}+Z-R_{1i})/D$ $\sigma_{12} = -R_{1i}/D,\ \sigma_{21} = R_{2i}/D$ $\sigma_{22} = R_{2i}/D,\ D = R_{2i}+Z$
	even	$\begin{matrix} 0 & -nR_{1i} \\ n & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & -nR_{1i} \\ -nR_{1i}G_{2i} & 0 \end{matrix}$	$\sigma_{11} = (G_{2i} + Y - G_{1i})/D$ $\sigma_{12} = -G_{1i}/D,\ \sigma_{21} = G_{2i}/D$ $\sigma_{22} = G_{2i}/D,\ D = Y+G_{2i}$
7	odd	$\begin{matrix} nG_{1i} & -n \\ 0 & -n \end{matrix}$	$\begin{matrix} -nG_{1i} & 0 \\ -nG_{1i} & -nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11} = \sigma_{11} = \sigma_{12} = R_{1i}/D$ $\sigma_{21} = -R_{2i}/D$ $\sigma_{22}=(R_{1i}-Z-R_{2i})/D,\ D = R_{1i}-Z$
	even	$\begin{matrix} -n & nR_{1i} \\ -n & 0 \end{matrix}$	$\begin{matrix} 0 & nR_{1i} \\ nR_{1i}G_{2i} & nR_{1i} \end{matrix}$	$\sigma_{11} = \sigma_{12} = G_{1i}/D$ $\sigma_{21} = -G_{2i}/D$ $\sigma_{22}=(G_{1i}-Y-G_{2i})/D,\ D=G_{1i}-Y$
8*	odd	$\begin{matrix} -nG_{1i} & -n \\ -nG_{1i} & 0 \end{matrix}$	$\begin{matrix} 0 & nG_{1i}R_{2i} \\ -nG_{1i} & nG_{1i}R_{2i} \end{matrix}$	$\sigma_{11} = 1,\ \sigma_{12} = -R_{1i}G_{2i}$ $\sigma_{21} = 1,$ $\sigma_{22} = G_{2i}[R_{2i}-Z-R_{1i}]$
	even	$\begin{matrix} -n & -nR_{1i} \\ 0 & -nR_{1i} \end{matrix}$	$\begin{matrix} -nR_{1i}G_{2i} & 0 \\ -nR_{1i}G_{2i} & nR_{1i} \end{matrix}$	$\sigma_{11}=1,\ \sigma_{12} = -G_{1i}R_{2i}$ $\sigma_{21}=1,$ $\sigma_{22} = R_{2i}[G_{2i}-G_{1i}-Y]$

NOTE: For i odd: $n = -R_{11}, \quad i = 1$

$$= -R_{1i} \prod_{j=1}^{(i-1)/2} [R_{1f}G_{1(f+1)}], \quad f = 2j \text{ otherwise}$$

For i even: $n = -1, \quad i = 2$

$$= - \prod_{j=1}^{(i-2)/2} [R_{1f}G_{1(f+1)}], \quad f = 2j \text{ otherwise}$$

TABLE 1 (continued)

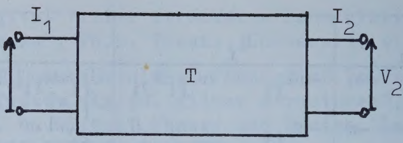


Fig. 1.

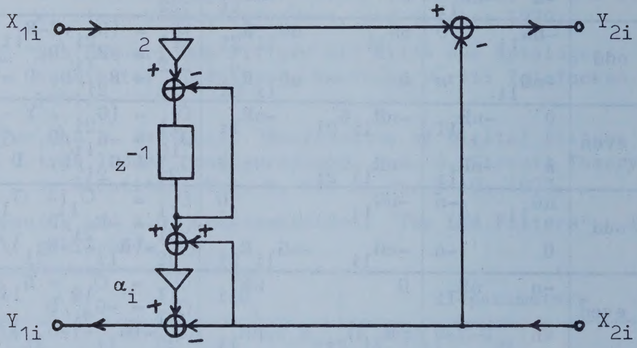

X1i
+
Y2i
-
2
+
-
z-1
+
+
αi
+
Y1i
-
X2i

Fig. 2.

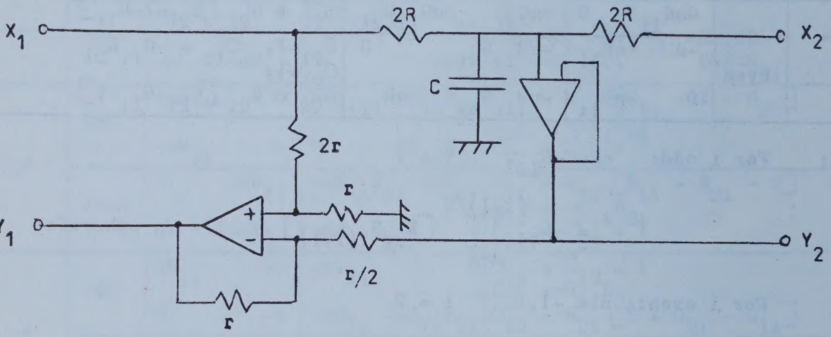

X1
2R
2R
X2
C
2r
r
Y1
+
-
r/2
r
Y2

Fig. 3.

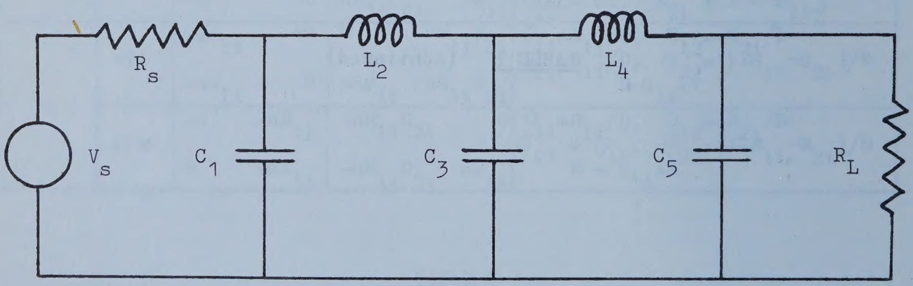

Rs
L2
L4
Vs
C1
C3
C5
RL

Fig. 4.

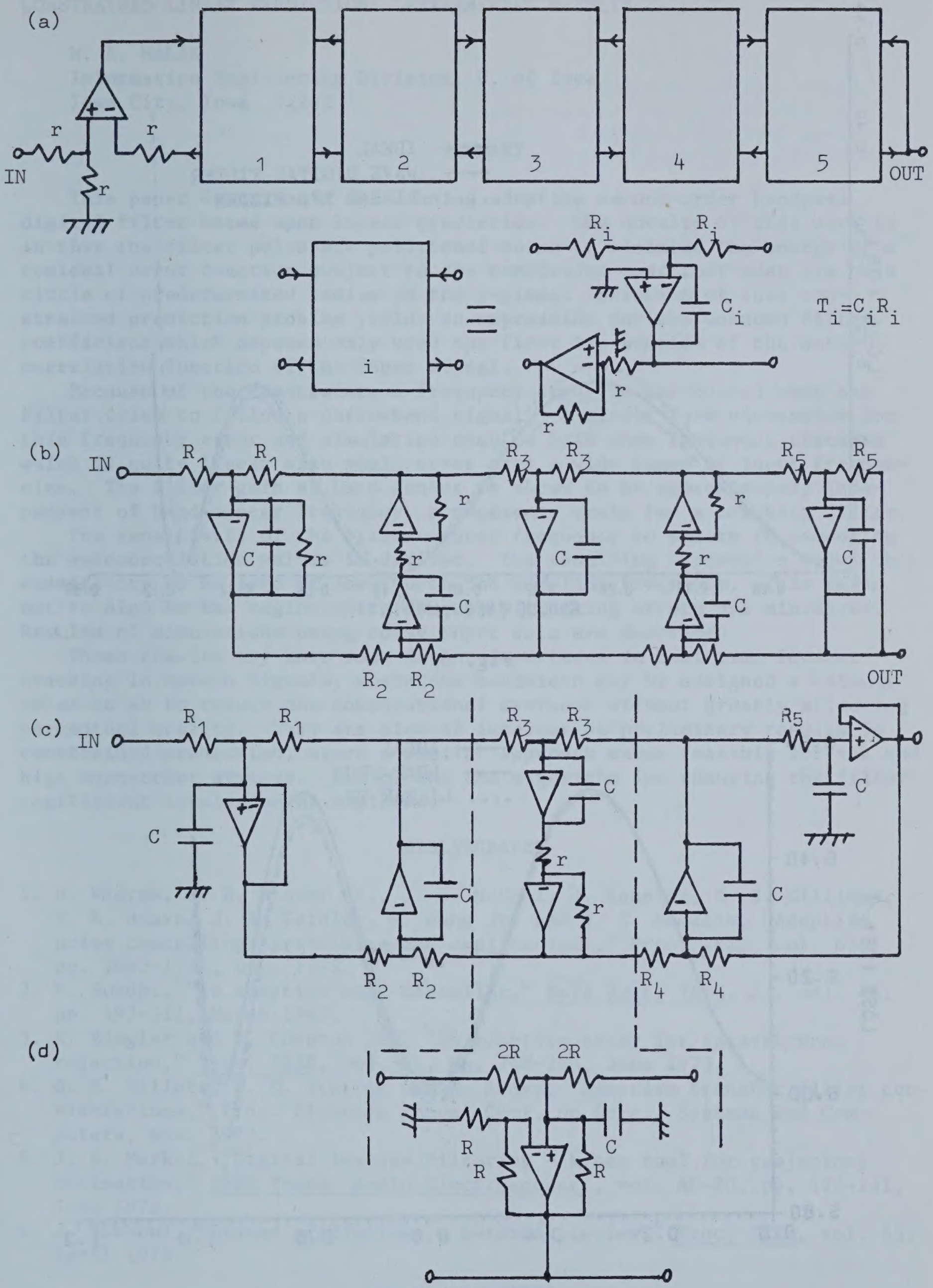

(a)
IN
r
r
r
1
2
3
4
5
OUT
i
R_i
R_i
C_i
r
r
T_i=C_iR_i
(b)
IN
R_1
R_1
r
C
r
r
C
R_3
R_3
C
r
r
C
R_5
R_5
C
R_2
R_2
OUT
(c)
IN
R_1
R_1
C
C
R_3
R_3
C
r
r
C
R_5
C
R_2
R_2
R_4
R_4
(d)
2R
2R
R
C
R
R

Fig. 5. (Note that when the non-inverting input of the OP AMP is not shown, it is assumed to be grounded.)

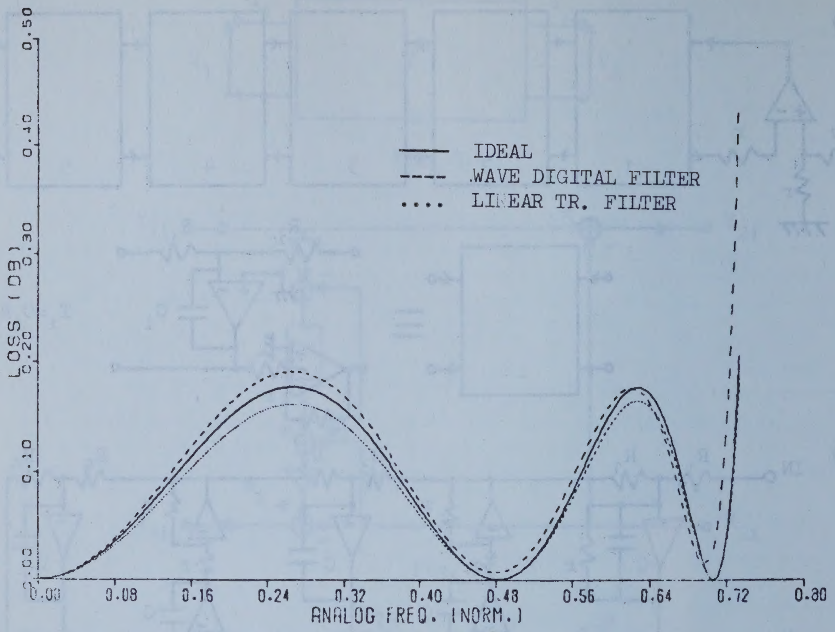

IDEAL
WAVE DIGITAL FILTER
LINEAR TR. FILTER
LOSS (DB)
0.00
0.10
0.20
0.30
0.40
0.50
0.00
0.08
0.16
0.24
0.32
0.40
0.48
0.56
0.64
0.72
0.80
ANALOG FREQ. (NORM.)

Fig. 6.

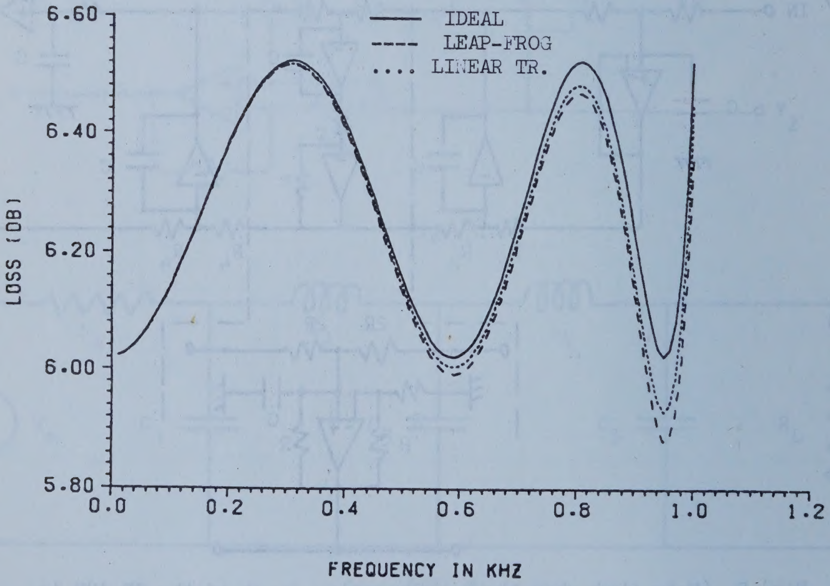

IDEAL
LEAP-FROG
LINEAR TR.
LOSS (DB)
6.60
6.40
6.20
6.00
5.80
0.0
0.2
0.4
0.6
0.8
1.0
1.2
FREQUENCY IN KHZ

Fig. 7.

CONSTRAINED LINEAR PREDICTION: PRELIMINARY RESULTS

N. R. MALIK
Information Engineering Division, U. of Iowa
Iowa City, Iowa 52242

SUMMARY

This paper describes a self-tuning adaptive second-order bandpass digital filter based upon linear prediction. The novelty of this work is in that the filter poles are positioned so as to minimize the energy of a residual error function subject to the constraint that they must lie on a circle of predetermined radius in the z-plane. Solution of this constrained prediction problem yields an expression for the unknown filter coefficient which depends only upon the first two samples of the autocorrelation function of the input signal.

Because of the constraint, a frequency error is introduced when the filter tries to follow a narrowband signal. A closed form expression for this frequency error and simulation results both show frequency tracking which is quite linear with small error over a wide range of input frequencies. The filter gain at band center is shown to be approximately independent of band-center frequency, a necessary trait for a tracking filter.

The sensitivity of the filter center frequency to errors in computing the autocorrelation values is derived. The resulting expression shows this sensitivity to be zero at one-fourth the sampling frequency. This turns out to also be the region where frequency tracking errors are minimized. Results of simulations using noisy input data are described.

These results may have some direct importance in real-time formant tracking in speech signals, where the bandwidth may be assigned a default value so as to reduce the computational overhead without greatly affecting perceptual quality. They are also of interest as preliminary results in constrained prediction, since a similar approach seems feasible for 4th and high even-order systems. Use of the LMS algorithm for changing the filter coefficient is also being explored.

BIBLIOGRAPHY

1. B. Widrow, J. R. Glover Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong Jr. and R. C. Goodlin, "Adaptive noise cancelling: principles and applications," Proc. IEEE, vol. 63, pp. 1692-1716, Dec. 1975.
2. M. Sondhi, "An adaptive echo canceller," Bell Syst. Tech. J., vol. 46, pp. 497-511, March 1967.
3. R. Riegler and R. Compton Jr., "An adaptive array for interference rejection," Proc. IEEE, vol. 61, pp. 748-758, June 1973.
4. G. R. Elliott, S. D. Stearns and N. Ahmed, "Adaptive transfer filter considerations," Proc. Eleventh Annual Conf. on Ccts., Systems and Computers, Nov. 1977.
5. J. D. Markel, "Digital inverse filtering - a new tool for trajectory estimation," IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 129-141, June 1972.
6. J. Makhoul, "Linear prediction: a tutorial review," Proc. IEEE, vol. 63, April 1975.

COMBINED SOURCE-CHANNEL CODING OF IMAGES†

J. W. MODESTINO AND D. G. DAUT
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

A combined source-channel coding approach is described for the encoding, transmission and remote reconstruction of image data. The source encoder employs two-dimensional (2-D) differential pulse code modulation (DPCM). This is a relatively efficient encoding scheme in the absence of channel errors. In the presence of channel errors, however, the performance degrades rapidly. By providing error control protection to those encoded bits which contribute most significantly to image reconstruction, it is possible to minimize this degradation without sacrificing transmission bandwidth. The result is a relatively robust design which is reasonably insensitive to channel errors and yet provides performance approaching the rate-distortion bound.

I. INTRODUCTION

There has been widespread interest recently in the efficient encoding of image data. Two-dimensional (2-D) differential pulse code modulation (DPCM) is a technique which has been widely employed for this purpose. Its behavior in the absence of channel errors has been studied extensively and is well-documented [1]-[5]. In the presence of channel errors, however, the behavior of 2-D DPCM degrades rapidly. As a result, some form of error control protection must be provided if high-quality image reconstruction is to be achieved. Since channel coding generally entails a bandwidth expansion, this operation can be extremely wasteful of channel bandwidth unless applied judiciously. In particular, tradeoffs must be made between the reconstruction accuracy associated with the source coder and the degree of error control protection provided by the channel coding.

In this paper we describe a combined source-channel coding approach for the encoding, transmission and remote reconstruction of image data which exploits these tradeoffs. The source encoder employs 2-D DPCM which has been appropriately matched to the image source. The result is a relatively robust design which is reasonably insensitive to channel errors and yet provides performance approaching an achievable rate-distortion bound.

II. PRELIMINARIES

While there are a wide range of possible stochastic models for images, we will restrict attention to the class of 2-D autoregressive random fields described according to

$$S_{i,j} = \sum_{k=0}^{K} \sum_{\ell=0}^{L}{}' a_{k,\ell} S_{i-k,j-\ell} + W_{i,j} \quad ; \quad i,j \geq 0 \qquad (1)$$

where the prime is intended to indicate that the point $k=\ell=0$ is excluded from the double summation and $\{W_{i,j}\}$ is a 2-D zero-mean sequence of independent and identically distributed (i.i.d.) random variables possessing common variance σ_w^2. We assume that the initial values $S_{-k,-\ell}$ for $k=0,1,\ldots,K-1$ and $\ell=0,1,\ldots,L-1$ have been specified.

As a concrete example of a 2-D autoregressive process we note the 2-D Gauss-Markov random field.

---

† This work supported in part by ONR under Contract No.N00014-75-C-0281.

Here

$$S_{i,j}=\rho_1 S_{i-1,j} + \rho_2 S_{i,j-1} - \rho_1\rho_2 S_{i-1,j-1} + W_{i,j} \tag{2}$$

with $0\leq|\rho_i|\leq 1$, $i=1,2$ and $\{W_{i,j}\}$ a 2-D i.i.d. zero-mean Gaussian sequence possessing common variance $\sigma_w^2 = \sigma_s^2 (1-\rho_1^2)(1-\rho_2^2)$. The quantity σ_s^2 represents the common variance of the resulting sequence $\{S_{i,j}\}$, i.e., $\sigma_s^2=\mathrm{var}\{S_{i,j}\}$, $i,j\geq 1$. We will use this 2-D Gauss-Markov random field as an illustrative example in what follows.

An overall block diagram of the DPCM system under consideration is provided in Fig. 1. Following established nomenclature, the DPCM coding/decoding scheme is said to be matched to the message model if $\alpha_{i,j}=a_{i,j}$. That is, the coefficients in the feedback predictor are chosen equal to the 2-D autoregressive coefficients in (1). For analysis purposes we assume this to be the case although there may be good reason for other choices.

In Fig. 1, the sequence $\{X_{i,j}\}$ represents the transmitted 2-D sequence whose values assume one of the $Q=2^n$ possible output levels of the n-bit quantizer. Exclusive use will be made of a symmetric uniform quantizer with step size Δ normalized to the standard deviation σ_e of the error sequence $\{E_{i,j}\}$. That is, the quantization boundaries $\{E_\ell\}$ and output levels $\{X_\ell\}$ satisfy

$$E_\ell - E_{\ell-1} = X_\ell - X_{\ell-1} = \Delta\sigma_e \quad ; \quad \ell=2,3,\ldots,Q-2 \tag{3}$$

while $E_0=-\infty$ while $E_Q=\infty$. The quantizer output level $X_\ell=[\ell-(Q-1)/2]\Delta\sigma_e$, $\ell=0,1,\ldots,Q-1$ is then expressed as an n-bit binary word for a subsequent transmission over the digital data link. In what follows we assume that a natural binary representation is employed. That is, the output level X_ℓ is coded into the natural binary representation of the integer ℓ. It will be assumed that individual bit streams are available for subsequent modulation/coding and transmission. In particular, the modulation/coding is allowed to be different for each of the n serial bit streams associated with the output of the n-bit quantizer.

In all cases, the quantizer characteristics are chosen to minimize the mean-square quantization error. That is, the single parameter Δ is chosen to minimize the quantity

$$\sigma_q^2 = E\{Q_{i,j}^2\} = \sum_{\ell=0}^{Q-1} \int_{E_\ell}^{E_{\ell+1}} [\xi - X_\ell]^2 p_E(\xi)d\xi \tag{4}$$

where $p_E(\cdot)$ is the probability density function (p.d.f.) associated with the error sequence $\{E_{i,j}\}$. It is extremely difficult to provide explicit evaluation of $p_E(\cdot)$ in any realistic situation. From the experimental results provided in [6] there is substantial evidence that the error sequence $\{E_{i,j}\}$ is approximately Laplacian distributed for a wide-range of real-world images. For the 2-D Gauss-Markov random field described by (2), on the other hand, we would expect the error sequence to be approximately Gaussian distributed. In what follows we consider only the case of an optimum Gaussian quantizer. Tabulations of the optimum Δ for various number of quantization levels Q have been provided for the Gaussian distribution in Max [7].

Finally the sequence $\{R_{i,j}\}$ appearing at the channel output is described according to

$$R_{i,j} = X_{i,j} + N_{i,j} \tag{5}$$

where the sequence $\{N_{i,j}\}$ represents additive digital channel noise assumed independent from pixel-to-pixel. The reproducing estimate $\{\hat{S}_{i,j}\}$ delivered to the destination is then determined recursively according to

$$\hat{S}_{i,j} = \sum_{k=0}^{K} {\sum_{\ell=0}^{L}}' \alpha_{k,\ell}\hat{S}_{i-k,j-\ell}+R_{i,j} \; ; \; i,j \geq 0 \tag{6}$$

again assuming that appropriate initial conditions have been specified. In the absence of channel errors (i.e., $R_{i,j}=X_{i,j}$) we have $\hat{S}_{i,j}=S_{i,j}+Q_{i,j}$, so that the only source of error is the quantization noise represented by $\{Q_{i,j}\}$.

In Fig. 2 we illustrate typical behavior of this 2-D DPCM encoding scheme on an outdoor scene for an optimum uniform Gaussian quantizer in the absence of channel errors. The image here, as in the remainder of the paper is 256 x 256 array. These results assume a first-order 2-D autoregressive image model similar to that described by (2). In particular, the order of the predictor $\hat{S}^{+}_{i,j}$ in Fig. 1 is such that K=L=1. Least-squares estimates of the vertical and horizontal correlation coefficients, ρ_1 and ρ_2 respectively, were obtained and the predictor feedback coefficients set equal to the corresponding estimated quantities.

In the absence of channel errors the 2-D DPCM encoding scheme provides reasonably good fidelity even for 1 bit/pixel although there are some distortions particularly in the vicinity of sharp edges. There is marginal subjective improvement beyond 2-3 bits/pixel. Results using the Laplacian quantizer provide barely perceptible subjective improvement indicating some insensitivity to quantizer choice.

The effects of channel errors can be observed in Fig. 3. Here the channel is modeled as a binary symmetric channel (BSC) with bit error probability $P_b=10^{-3}$. Subjective tests (cf. [8]) have indicated a sharp threshold at this point with reconstructed image quality degrading rapidly beyond this point. The effects of channel errors are much more pronounced as the number of quantization levels Q increases. Indeed, for these channel error rates it would appear wasteful of channel bandwidth to employ more than 1-2 bits/pixel. Increasing the quantizer accuracy degrades rather than improves image quality. The explanation for this behavior can be seen most clearly from the explicit evaluation of output signal-to-noise ratio provided in the following section.

III. ANALYSIS OF OUTPUT SIGNAL-TO-NOISE RATIO

The output signal-to-noise ratio (SNR_0) of the 2-D DPCM system is given by

$$SNR_0 = \frac{(\sigma_s^2/\sigma_e^2)}{e_T^2} \tag{7}$$

where σ_s^2 is the variance of the assumed zero-mean stationary 2-D sequence $\{S_{i,j}\}$ and

$$e_T^2 = E\{[S_{i,j}-\hat{S}_{i,j}]^2\}/\sigma_e^2 \tag{8}$$

is the resulting mean-square reconstruction error normalized to the variance σ_e^2 of the stationary sequence $\{E_{i,j}\}$ appearing as input to the quantizer. The quantity (σ_s^2/σ_e^2) has an interpretation as the mean-square signal level normalized to the mean-square prediction error which, if we neglect the effects of quantization noise, is easily shown to be given by

$$(\sigma_s^2/\sigma_e^2) = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |D(e^{j\lambda_1},e^{j\lambda_2})|^2 d\lambda_1 d\lambda_2 \tag{9}$$

Here $D(z_1,z_2)$ is defined as a function of the complex variables z_1,z_2 as

$$D(z_1,z_2) = \frac{1}{1-H(z_1,z_2)} \tag{10}$$

where $H(z_1,z_2)$ is the transfer function of the feedback predictor in the encoder. For example, in the case of the 2-D Gauss-Markov process in (2) we have

$$(\sigma_s^2/\sigma_e^2) = \frac{1}{(1-\rho_1^2)(1-\rho_2^2)} \tag{11}$$

Following a fairly standard development [9]-[12], the quantity e_T^2 in (7) can be expressed as the sum of three separate and conveniently normalized components according to

$$e_T^2 = \varepsilon_q + 2\varepsilon_m + \varepsilon_c \ . \tag{12}$$

Here, ε_q represents the mean-square value of the quantization noise normalized to σ_e^2. Similarly, the quantity ε_c represents the normalized mean-square error contribution due solely to channel errors. Finally the quantity ε_m represents a normalized mutual error term. Evaluation of these terms is given in [8] from which we will draw heavily.

As stated previously, the quantizer has been designed to minimize the variance σ_q^2 of the quantization noise. It is easily shown that

$$\varepsilon_q = 2 \sum_{\ell=1}^{Q/2-1} \int_{(\ell-1)\Delta}^{\ell\Delta} [y-(\ell-1/2)\Delta]^2 p_E(y)dy + 2\int_{(Q/2-1)\Delta}^{\infty} [y-(Q-1)\Delta/2]^2 \hat{p}_E(y)dy \tag{13}$$

where $\hat{p}_E(y) \triangleq \sigma_e p_E(\sigma_e y)$ is a normalized version of the p.d.f. associated with the error sequence possessing zero mean and unit variance. This expression for ε_q is easily evaluated numerically for various values of Q and Δ.

Consider now the error term ε_c contributed by channel error effects. Assuming that the channel noise sequence $\{N_{i,j}\}$ is independent from pixel-to-pixel with mean-square value e_n^2 we have (cf. [8])

$$\varepsilon_c = (e_n^2/\sigma_e^2)(\frac{1}{2\pi})^2 \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |D(e^{j\lambda_1},e^{j\lambda_2})|^2 d\lambda_1 d\lambda_2 \tag{14}$$

where $D(z_1,z_2)$ is the discrete 2-D system transfer function associated with the decoder and

$$e_n^2 = E\{N_{i,j}^2\} = \sum_{k=0}^{Q-1} \sum_{\ell=0}^{Q-1} (X_k-X_\ell)^2 P_{k|\ell} P_\ell \ . \tag{15}$$

This is merely the mean-square error incurred if X_ℓ was transmitted and decoded as the level X_k averaged over the probability of all such error events. The quantity $P_{k|\ell}$ is the conditional probability that level X_ℓ was transmitted and decoded as level X_k while P_ℓ is the probability of transmitting level X_ℓ. It is more illuminating to consider the normalized quantity e_n^2/σ_e^2 which is easily seen to reduce to

$$e_n^2/\sigma_e^2 = \Delta^2 \sum_{k=0}^{Q-1} \sum_{\ell=0}^{Q-1} (k-\ell)^2 P_{k|\ell} P_\ell \ . \tag{16}$$

In [8] it is shown that this last quantity can be evaluated according to

$$e_n^2/\sigma_e^2 = \Delta^2 \{ \sum_{i=0}^{n-1} P_{b_i}(1-P_{b_i})2^{2i} + [\sum_{i=0}^{n-1} P_{b_i}(1-P_{b_i})2^i]^2 \} , \tag{17}$$

where P_{b_i}, i=0.1.,,,.n-1 is the bit error probability associated with each of the successive bit positions in the n-bit quantizer output word. For the particular case of equal bit error probability $P_{b_i}=P_b$, $i=0,1,\ldots,n-1$ and this last expression reduces to

$$e_n^2/\sigma_e^2 = \Delta^2 P_b(1-P_b)\{(Q^2-1)/3 + P_b(1-P_b)(Q-1)^2\} , \tag{18}$$

where we recall $Q=2^n$. In either case, the error contribution ε_c is readily evaluated for various values of Q and Δ. Again in the particular case of the 2-D Gauss-Markov process given by (2) we find

$$\varepsilon_c = \frac{e_n^2/\sigma_e^2}{(1-\rho_1^2)(1-\rho_2^2)} \qquad (19)$$

with e_n^2/σ_e^2 given by either (17) or (18) as appropriate.

Finally, the mutual error term ε_m is shown in [8] to be expressible in the form

$$\varepsilon_m = 2\Delta\Big[\sum_{\ell=Q/2}^{Q-1}\Big\{\sum_{i=0}^{n-1}(1-2\ell_i)P_{b_i}2^i\Big\}\varepsilon'_\ell\Big]\Big(\frac{1}{2\pi}\Big)^2\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}D(e^{j\lambda_1},e^{j\lambda_2})d\lambda_1 d\lambda_2. \qquad (20)$$

Here ℓ_i, i=0,1,...,n-1 represent coefficients of the binary expansion of the integer ℓ according to

$$\ell = \sum_{i=0}^{n-1}\ell_i 2^i \quad ; \ \ell_i = 0,1 \qquad (21)$$

while P_{b_i} is the bit error probability associated with the transmission of the i'th bit and finally

$$\varepsilon'_\ell = \int_{E_\ell/\sigma_e}^{E_{\ell+1}/\sigma_e}\{y-[\ell-(Q-1)/2]\Delta\}\hat{p}_E(y)dy \quad ; \ \ell=0,1,\ldots,Q-1 \qquad (22)$$

is the average normalized quantization error given that the ℓ'th level was transmitted. For the particular case where the bit error probability is constant for each bit position we have

$$\varepsilon_m = -4\Delta P_b\Big[\sum_{\ell=Q/2}^{Q-1}\{\ell-(Q-1)/2\}\varepsilon'_\ell\Big]\Big(\frac{1}{2\pi}\Big)^2\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}D(e^{j\lambda_1},e^{j\lambda_2})d\lambda_1 d\lambda_2 \ . \qquad (23)$$

In either case, the quantity ε_m is readily computed as a function of Q and the channel error probability for various image models.

For the particular case of the 2-D Gauss-Markov sequence described by (2) it is readily seen that

$$\Big(\frac{1}{2\pi}\Big)^2\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}D(e^{j\lambda_1},e^{j\lambda_2})d\lambda_1 d\lambda_2 = 1 \qquad (24)$$

which simplifies evaluation of ε_m. This component has been found to be negligible compared to the other error components ε_q and ε_c.

In Fig. 4 we illustrate the behavior of SNR_0 for an assumed 2-D Gauss-Markov image model of the outdoor scene. Here a Gaussian uniform quantizer was used and individual bits were transmitted using coherent binary phase-shift keyed (BPSK) modulation over an additive white Gaussian noise (AWGN) channel. The bit error probability in this case is given by

$$P_{b_i} = Q\left(\sqrt{\frac{2E_{si}}{N_0}}\right) \quad ; \ i=0,1,\ldots,n-1 \qquad (25)$$

where

$$Q(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^{\infty}e^{-y^2/2}dy \qquad (26)$$

and E_{si} is the signal energy per transmitted binary channel symbol while $N_0/2$ is the double-sided noise spectral density in watts/Hz.

In this figure, the quantity SNR_0 is plotted as a function of the channel signal-to-noise ratio on a normalized per pixel basis. That is, if N uses of the channel are employed to transmit the n-bit quantizer output word corresponding to each pixel position then the channel signal-to-noise ratio is defined according to

$$\mathrm{SNR}_i = \sum_{i=0}^{N-1} (2E_{si}/N_0) \qquad (27)$$

where E_{si}, i=0,1,...,N-1 is the signal energy for each channel use. Actually, in Fig. 4, N=n while we assume constant energy so that $E_{si}=E_s$, i=0,1, ...,n-1. It will prove convenient when employing channel coding to allow the generality afforded by (27). The bandwidth expansion factor is then N/n.

Also included in Fig. 4 is the rate-distortion bound on SNR_0 for operation over an AWGN channel. This is obtained by equating the rate-distortion function, computed under a mean-square fidelity criterion, to the channel capacity and solving for SNR_0 as a function of SNR_i. The rate-distortion function R(D) for a continuous-amplitude 2-D Gaussian sequence has been given by Stuller and Kurz [13] following the development by Berger [14] in the 1-D case. A computer program for the numerical evaluation of R(D) has been constructed as described in [15]. In the present context, we have $\mathrm{SNR}_0=\sigma_s^2/D$ where D is the mean-square distortion. The capacity of the AWGN channel with signal energy E_{si} is given by the well-known expression [16]

$$C = \frac{1}{2}\log_2[1+\frac{2E_{si}}{N_0}] \text{ (bits/channel use)} \qquad (28)$$

so that equating R(D) to C allows solution for SNR_0 in terms of SNR_i.

Several things should be observed from Fig. 4. In the first place, note the sharp threshold effect which becomes more pronounced as the number of quantization levels Q increases. For SNR_i in excess of approximately 20 dB channel errors are rare and the performance is limited solely by quantization noise. An increase in the number of quantization levels results in a commensurate increase in SNR_0. For smaller SNR_i we see a complete reversal of this behavior. For example, at SNR_i=12dB relative performance, measured in terms of SNR_0, actually improves by decreasing the number of quantization levels. This is an illustration of the behavior noted previously. It should be noted that this threshold behavior occurs at error probabilities typical of that provided on many commercial and military data links.

The second observation to be made from Fig. 4 is that this 2-D DPCM scheme is relatively inefficient compared to the rate-distortion bound. In the next section we develop ultimate performance bounds to be achieved through channel coding.

IV. THEORETICAL PERFORMANCE BOUNDS

A useful upper bound on SNR_0 can be obtained by lower bounding the bit error probability P_b ,i=1,2,...,n-1 associated with each of the bits in the n-bit quantizer output word. Consider first the case where the modulation/coding system treats each bit identically. Recall that N uses of the channel are required to transmit the quantizer output word so that the normalized code rate in bits per channel use is R=n/N. Under the AWGN assumption, the channel capacity is given by (28) with $E_{si}=E_s$, the constant energy per channel use. The converse to the coding theorem (cf. [16], [17]) then provides the desired lower bound on the common bit error probability P_b. Following the particularly illuminating development in McEliece [18], at rates above the capacity C of the AWGN channel, the bit error probability achievable by any rate R code is bounded away from zero by

$$P_b \geq H^{-1}(1-C/R) \qquad (29)$$

where $H^{-1}(\cdot)$ represents the inverse of the binary entropy function

$$H(x) = - x \log_2 x - (1-x)\log_2(1-x);\ 0<x<1 \tag{30}$$

At rates below capacity P_b can be made negligibly small. For purposes of numerical evaluation, P_b will be computed according to the right-hand side of (29) for $R>C$ and we set $P_b=0$ for $R\leq C$. It should be noted that this approach is similar to that employed by Chase et al.,[19] who observe that no known codes can achieve this lower bound on P_b and even for $R<C$ infinitely long codes are required to achieve arbitrarily small bit error probability. Nevertheless, as we shall see, this bound does provide a useful perspective in assessing the efficacy of combined source-channel coding.

In Fig. 5 we illustrate typical performance results for the outdoor scene employing a fixed-rate modulation/coding approach with $N=5$ channel transmissions per pixel. Fixed-rate $R=n/N$ for $n=1,2,\ldots,4,5$ codes then insure common transmission bandwidth requirements. These curves indicate clearly the nature of the tradeoffs possible between quantization accuracy and sensitivity to channel errors at a fixed transmission bandwidth. The coded system performance is characterized by a reduction in saturation level SNR_0 but with a threshold occurring at significantly lower SNR_i. Indeed, the knee of this threshold characteristic occurs in all cases within 2-3dB of the rate-distortion bound.

While the fixed-rate coding approach appears to offer performance approaching the rate-distortion bound, it should be noted that codes of fairly unusual rates like $R=2/5$, $3/5$, $4/5$, etc. are required. Relatively little is known concerning the construction and properties of codes of these rates. It remains to demonstrate the degree to which these ultimate performance bounds can be approached by practical coding schemes. In particular, we restrict attention to short constraint length convolutional codes of rates $R=1/2$, $1/3$ and $1/4$ decoded using the Viterbi algorithm, an excellent description of which can be found in [20]. Restriction to these rates necessitates unequal protection on each bit. The optimum binary codes tabulated by Odenwalder [21] and Larsen [22] are used exclusively in what follows.

Typical results are illustrated in Fig. 6 for the case $N=3$ channel uses per pixel where optimum codes of constraint length $K=3$, 6 and 9 have been employed. Here we have employed tight exponential bounds on P_b (cf. [8] for detials). The convergence of the practical code performance toward the optimum performance with increasing constraint length K is illustrated clearly. Unfortunately, the performance even for a $K=9$ code falls short of the optimum code performance by some 2-3dB and slightly more when compared to the rate-distortion bound.

Also included in Fig. 6 is a dotted curve labeled R_0^* bound which is displaced some 2-3dB from the rate-distortion bound. We feel that this quantity provides a more realistic bound on achievable performance than the rate-distortion bound. In particular, it is well-known (cf. [11],[23]) that the critical rate R_0 provides a realistic bound on the rates achievable for a reasonable degree of code complexity. This is true for either block or convolutional codes. Furthermore, the quantity (cf. [23])

$$R_0^* = \frac{\log_2 e}{2}\left[1+E_s/N_0-\sqrt{1+(E_s/N_0)^2}\right]+ \frac{1}{2} \log_2\left[\frac{1}{2}\left(1+\sqrt{1+(E_s/N_0)^2}\right)\right] \tag{31}$$

provides a useful upper bound on R_0 on the AWGN channel for any choice of modulator-demodulator parameters. The parameter $R_0^*<C$ then provides a more realistic limit than the channel capacity C on the range of rates achievable on the AWGN channel. In Fig. 6 the curve labeled R_0^* bound then is obtained by equating R_0^* to the rate-distortion function $R(D)$ and solving for SNR_0 as a function of SNR_i. We see then that the use of practically implemented

short constraint length convolutional codes results in performance within 1-2dB of the "practically achievable" R_0^* bound. Simulation results reported in [8] have substantiated these conclusions.

V. SUMMARY AND CONCLUSIONS

We have described an approach to the combined source-channel coding of a particular class of image sources. This approach offers theoretical performance approximating the rate distortion bound. We have shown how this theoretical performance can in turn be approximated by a class of practical short constraint length convolutional codes and demonstrated the performance gains through selected simulations on real-world images. The major deficiency likes in the lack of good high rate code constructions required to achieve the near optimum behavior indicated by the fixed-rate code performance results. More specifically, we have been forced to utilize previously tabulated optimum codes for rates R=1/2, 1/3 and 1/4. This severely compromises the flexibility of the approach in those situations where extremely high quality image reconstruction is required, i.e., high saturation level SNR_0 at low SNR_i. For more moderate reconstructed image quality requirements this restriction to available codes does not present a problem. The result then, at least for moderate reconstructed image quality requirements, is a relatively robust design which offers a near optimum compromise between quantization accuracy and sensitivity to channel errors while preserving a fized transmission bandwidth requirement.

REFERENCES

1. D. F. Connor, R.C.Brainard and J.O.Limb, "Intraframe Coding for Picture Transmission", Proceedings IEEE, vol. 60, pp. 779-791, July 1972.

2. J. B. O'Neal, "Predictive Quantizing System (Differential Pulse Code Modulation) for the Transmission of Television Signals", Bell Syst. Tech.J. vol. 45, pp. 689-721, May-June 1966.

3. D. J. Connor, R.F.W. Pease and W.G.Scholes, "Television Coding Using Two-Dimensional Spatial Prediction", Bell Syst. Tech.J., vol. 50, pp.1049-1061, March 1971.

4. J. O. Limb and F.W.Mounts, "Digital Differential Quantizer of Television" Bell. Syst. Tech.J., vol. 48, pp. 2583-2599, September 1969.

5. J. O. Limb, C.R.Rubinstein and K.A.Walsh, "Digital Coding of Color Picturephone Signals by Element-Differential Quantization", IEEE Trans. Commun., vol. COM-19, pt. 1, pp. 992-1005, December 1971.

6. D. G. Daut, "An Empirical Study of Two-Dimensional Differential Pulse Code Modulation Encoding of Images", M. S. Thesis, Electrical and Systems Engineering Dept., Rensselaer Polytechnic Institute, Troy, N.Y., July 1977.

7. J. Max, "Quantizing for Minimum Distortion", IEEE Trans.Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

8. J. W. Modestino and D.G.Daut, "Combined Source-Channel Coding of Images", submitted to IEEE Trans. on Communication.

9. R. E. Totty and G.C.Clark, "Reconstruction Error in Waveform Transmission", IEEE Trans. Inform. Theory, vol. IT-13, pp. 336-338, April 1967.

10. K.-Y. Chang and R.W. Donaldson, "Analysis, Optimization and Sensitivity Study of Differential PCM Systems Operating on Noisy Communications Channels", IEEE Trans. Commun., vol. COM-20, pp. 338-350, June 1972.

11. J. E. Essman and P.A.Wintz, "The Effects of Channel Errors in DPCM Systems and Comparison with PCM Systems", IEEE Trans. Commun., vol. COM-21, pp. 867-877, August. 1973.

12. K.-Y Chang and R.W.Donaldson, "Nonadaptive DPCM Transmission of Monochrome Pictures Over Noisy Communication Channels", IEEE Trans. Commun., vol. COM-24, pp. 173-183, February 1976.

13. J. A. Stuller and B. Kurz, "Intraframe Sequential Picture Coding", IEEE Trans. Commun., vol. COM-25, pp. 485-495, May 1977.

14. T. Berger, Rate Distortion Theory: A Mathematical Basis for Data Compression, Prentice-Hall, Englewood Cliffs, N.J., 1971.

15. D. G. Daut, "Rate-Distortion Function for 2-D Image Model", unpublished Rensselaer Polytechnic Institute report, April 1978.

16. R. G. Gallager, Information Theory and Reliable Communication, Wiley, New York, 1968.

17. F. Jelinek, Probabilistic Information Theory, McGraw-Hill, New York, 1968.

18. R. J. McEliece, The Theory of Information and Coding, vol. 3, Encyclopedia of Mathematics and Its Applications, Addison-Wesley, Reading, MA., 1977.

19. D. Chase, et al., "Multi-Sample Error Protection Modulation Study", TR-76-161, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, New York, May 1976.

20. A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems", IEEE Trans. Commun., COM-19, pp. 751-772, October 1971.

21. J. P. Odenwalder, "Optimum Decoding of Convolutional Codes", Ph.D. Dissertation, Syst. Sci. Dept., Univ. California, Los Angeles, 1970.

22. K. J. Larsen, "Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4", IEEE Trans. Inform. Theory, vol. IT-19, pp. 371-372, May 1973.

23. J. M. Wozencraft and I.M.Jacobs, Principles of Communications Engineering, Wiley, New York 1965.

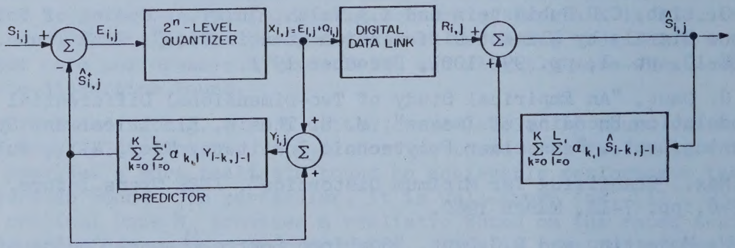
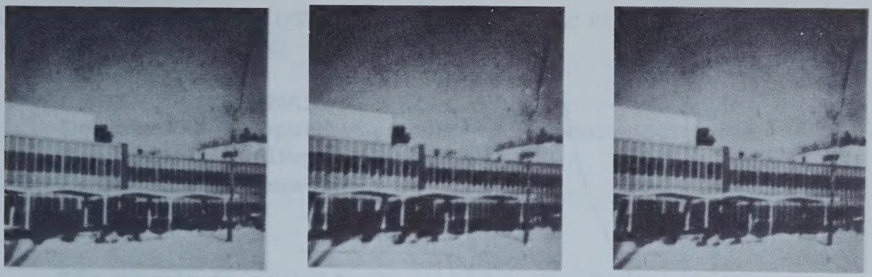

$S_{i,j}$
+
−
Σ
$E_{i,j}$
$\hat{S}_{i,j}$
2^n-LEVEL QUANTIZER
$X_{i,j}=E_{i,j}+Q_{i,j}$
DIGITAL DATA LINK
$R_{i,j}$
+
+
Σ
$\hat{S}_{i,j}$
$\sum_{k=0}^{K}\sum_{l=0}^{L}{}' a_{k,l}\, Y_{i-k,j-l}$
PREDICTOR
$Y_{i,j}$
+
Σ
+
$\sum_{k=0}^{K}\sum_{l=0}^{L}{}' a_{k,l}\, \hat{S}_{i-k,j-l}$

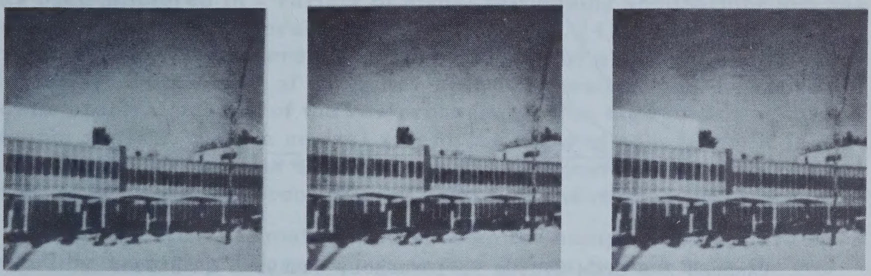
Figure 1

Block Diagram of Two-Dimensional DPCM

Encoding/Decoding System

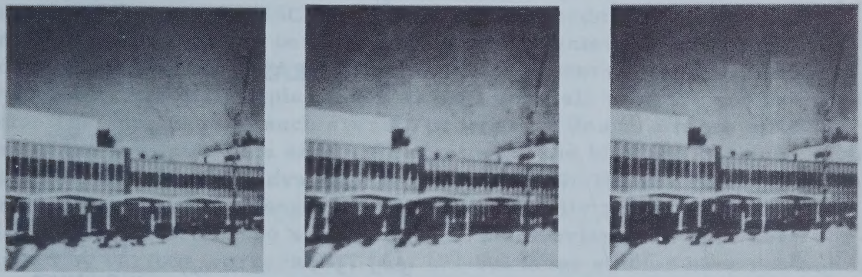


a) Original b) 1-bit DPCM c) 2-bit DPCM

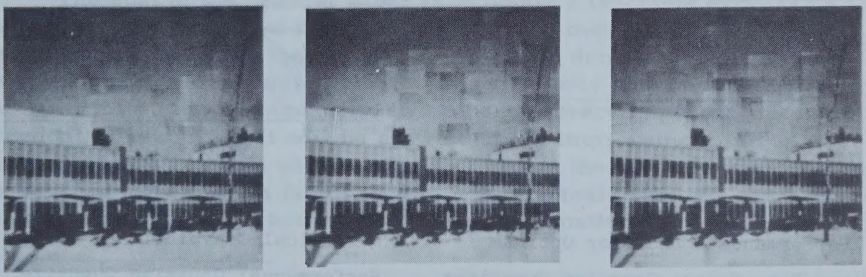


d) 3-bit DPCM e) 4-bit DPCM f) 5-bit DPCM

Figure 2 - 2-D DPCM Encoded Image; Gaussian Quantizer



a) Original b) 1-bit DPCM c) 2-bit DPCM



d) 3-bit DPCM e) 4-bit DPCM f) 5-bit DPCM

Figure 3 - 2-D DPCM Encoded Image; Gaussian Quantizer; $P_b=10^{-3}$

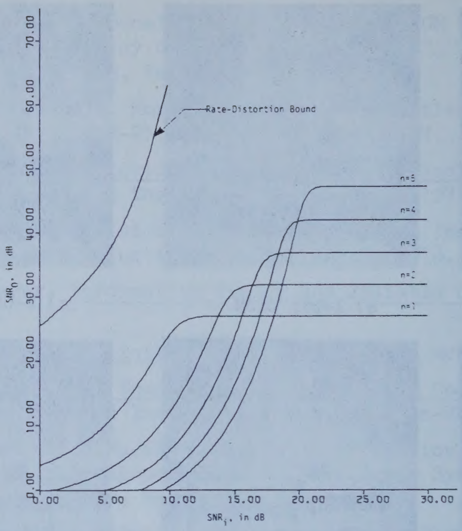


Figure 4

Uncoded 2-D DPCM System Performance; Gaussian Quantizer

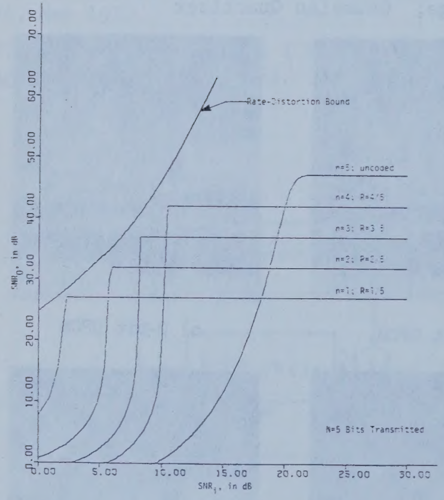


Figure 5

Performance Bounds for Optimum Fixed-Rate Codes; Gaussian Quantizer

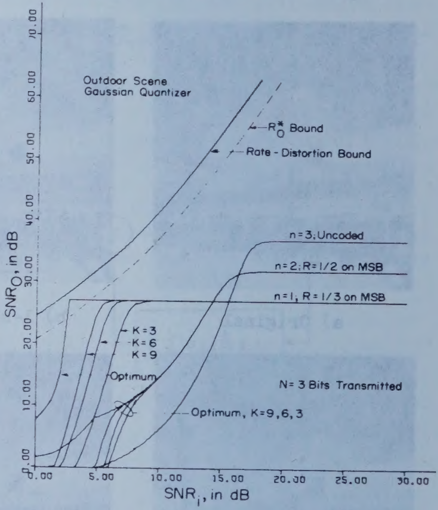


Figure 6

Practical Convolutional Code Performance; Gaussian Quantizer

ASYMPTOTIC DISTRIBUTION OF THE SAMPLE COHERENCE USING OVERLAP PROCESSING

ROBERT LUGANNANI
Department of Applied Physics and Information Science
University of California, San Diego
La Jolla, California 92093

INTRODUCTION

The magnitude-squared coherence function (MSC) provides a measure of the linearity of the relationship between two random processes. It has been employed in a variety of signal processing applications and has been the subject of several studies [1], [2], [3], [4]. In practice the magnitude-squared coherence is approximated by a sample statistic formed from estimates of the power spectral densities and the cross power spectral density of the two processes. The distribution of this sample statistic plays a central role in the study of coherence; it is needed, for example, to evaluate system performance measures such as probability of false alarm, probability of detection and mean-square error.

The spectral estimates used to form the sample statistic are obtained by averaging discrete Fourier transforms formed from the two processes. These transforms are constructed from samples of the processes that are spaced uniformly throughout successive observation intervals. When the processes are Gaussian and when the transforms formed from successive observation intervals are independent, the probability density of the sample MSC has been given by Goodman [5]. It is often desirable, however, to permit the observation intervals to overlap one another. In this case the transforms formed from successive intervals share one or more sample values and, in general, they are not independent. The advantage of such overlap processing lies in a more efficient use of the available data and in a reduction of the bias and variance of the sample MSC. The disadvantage is that the known results concerning the probability density no longer apply and the density remains unknown. Simulation studies of the sample MSC in the overlap case have been conducted by various workers [2], [6], [7] but these studies have not resulted in useful expressions for the density.

The goal of the present paper is to obtain a tractable expression for the probability density of the sample MSC when overlap processing is used. It will be shown that in certain cases the density obtained with overlap can be closely approximated by the density obtained without overlap but with a reduced number of independent observations (degrees of freedom). In addition it will be shown that the improvement in the bias and variance obtainable with overlap is limited; the bias and variance can be reduced at most by a factor of 2/3.* Numerical examples are presented to illustrate the accuracy of the approximations employed.

---

*A special case occurs when the two processes are uncorrelated. In this instance the appropriate factor for the reduction of the variance is 4/9.

PRELIMINARIES

Magnitude-Squared Coherence

Let $x_1(t)$ and $x_2(t)$ be real, zero-mean Gaussian processes. We assume that the processes are jointly wide sense stationary with correlation functions

$$R_{jk}(\tau) = E\, x_j(t+\tau)\, x_k(t), \qquad j,k = 1,2 \tag{1}$$

and associated power spectral densities

$$S_{jk}(f) = \int_{-\infty}^{\infty} R_{jk}(\tau) \exp[-2\pi i f \tau]\, d\tau, \quad j,k = 1,2\,. \tag{2}$$

The magnitude-squared coherence function is defined as the ratio

$$\gamma^2 = \gamma^2(f) = \frac{|S_{12}(f)|^2}{S_{11}(f) \cdot S_{22}(f)}\,. \tag{3}$$

It is readily verified that $0 \le \gamma^2 \le 1$ with $\gamma^2 = 0$ when $x_1(t)$ and $x_2(t)$ are uncorrelated and $\gamma^2 = 1$ when one of the processes can be expressed as a linear time-invariant operation on the other.

An important property of the MSC is its invariance under linear operations. Suppose the processes $y_1(t)$ and $y_2(t)$ are obtained via linear time-invariant operations on $x_1(t)$ and $x_2(t)$. Then from (3) it is evident that the MSC for $y_1(t)$ and $y_2(t)$ is identical to the MSC for $x_1(t)$ and $x_2(t)$. The extent to which the two MSC functions differ is a measure of either the nonlinearity or the nonstationarity of the relationship between the processes. Additional properties and applications of the MSC can be found in [1], [2], [3], [4].

The Sample Statistic

We wish to obtain an estimate of γ^2 from observations on $x_1(t)$ and $x_2(t)$. The processes will be sampled at uniformly spaced intervals and the sample values used to form the discrete Fourier transforms

$$X_j(f_r; n) = \Delta \sum_{k=0}^{K-1} x_j(k\Delta + [n-1]M\Delta) \exp[-2\pi i f_r \Delta k]\,,\ j = 1,2;\ n = 1,2,\ldots\,. \tag{4}$$

Here Δ is the sampling interval and the frequencies f_r are restricted to the values $f_r = r/M\Delta$, $r = 1, 2, \ldots, M-1$. The individual transforms are based on observations of the processes that are K samples long. Each observation interval is shifted by M samples $(M \le K)$ relative to the previous interval so that there is an overlap of $K-M$ samples in successive transforms. The restriction on the frequencies f_r is not a severe one in most applications; however, it greatly simplifies the analysis and allows us to concentrate on the salient features of the problem.

The transforms (4) are used to form estimates of the power spectral densities

$$\hat{S}_{jk}(f_r) = \frac{1}{NK\Delta} \sum_{n=1}^{N} X_j(f_r; n)\, X_k^*(f_r; n), \quad j, k = 1, 2 \tag{5}$$

and these, in turn, are used to form the sample MSC

$$\hat{\gamma}^2 = \hat{\gamma}^2(f_r) = \frac{|\hat{S}_{12}(f_r)|^2}{\hat{S}_{11}(f_r) \cdot \hat{S}_{22}(f_r)} \; . \tag{6}$$

In the following it will be convenient to rewrite (6) as

$$\hat{\gamma}^2 = \frac{\xi_3^2 + \xi_4^2}{\xi_1 \xi_2} \tag{7}$$

where $\xi_1 = \hat{S}_{11}(f_r)$, $\xi_2 = \hat{S}_{22}(f_r)$, $\xi_3 = \text{Real}\{\hat{S}_{12}(f_r)\}$ and $\xi_4 = \text{Imag}\{\hat{S}_{12}(f_r)\}$.

The problem addressed here is the determination of the probability density of $\hat{\gamma}^2$. If there is no overlap ($M = K$) and if the transforms $X_j(f_r; n)$ and $X_k(f_r; m)$ are independent whenever $n \neq m$, the density has been derived by Goodman [5]. It appears as

$$p(\hat{\gamma}^2) = (N-1)(1-\rho^2)^N (1-\hat{\gamma}^2)^{N-2} \; {}_2F_1(N, N, 1, \hat{\gamma}^2\rho^2) \tag{8}$$

where

$$\rho^2 = \frac{(E\,\xi_3)^2 + (E\,\xi_4)^2}{(E\,\xi_1) \cdot (E\,\xi_2)} \tag{9}$$

and ${}_2F_1$ is a hypergeometric function.

Formulation in Terms of Nonoverlapping Observations

We wish to rewrite (4) as a sum of other discrete Fourier transforms formed from nonoverlapping observation intervals. To do this each interval of length K is assumed to be an integer multiple of the shift M. That is, we assume that there exists an integer Q such that $K = MQ$. This is not a particularly restrictive assumption in most applications. Next define the discrete Fourier transforms

$$V_j(f_r; q) = \Delta \sum_{k=0}^{M-1} x_j(k\Delta + [q-1]M\Delta) \exp[-2\pi i f_r \Delta k], \; j = 1, 2; \; q = 1, 2, \ldots \; . \tag{10}$$

and note that these transforms are formed from nonoverlapping observations of the original processes. It is readily seen that $X_j(f_r; n)$ can be expressed as

$$X_j(f_r; n) = \sum_{q=0}^{Q-1} V_j(f_r; n+q), \quad j = 1, 2;\; n = 1, 2, \ldots \quad . \tag{11}$$

In general the transforms (10) associated with different observation intervals will be correlated. However, in many situations the correlation functions $R_{jk}(\tau)$ decrease rapidly enough so that the correlation between $V_j(f_r; p)$ and $V_k(f_r; q)$ can be neglected when $p \neq q$. We shall assume this situation obtains in the following and that the successive transforms are uncorrelated. Moreover, since $V_j(f_r; p)$ and $V_k(f_r; q)$ are Gaussian, the fact that they are uncorrelated for $p \neq q$ implies that they are independent.

The Gaussian Approximation

The joint probability density of the variables $\xi_1 - \xi_4$ is very complicated, and its determination is a formidable task. Nevertheless, this density (or an equivalent representation such as its characteristic function) must be known in order to find the density of $\hat{\gamma}^2$. In the following we shall simplify the problem greatly by making the assumption that, for sufficiently large N, the joint distribution of $\xi_1 - \xi_4$ is approximately Gaussian. The basis for this assumption is the central limit theorem. We do not prove the limit theorem here but simply note that it holds when there is no overlap [8] and observe that with overlap the desired result can be based on the assumed independence of the transforms $V_j(f_r; q)$.

To distinguish the true sample MSC from the approximate sample MSC obtained under the Gaussian assumption, we will denote the latter by z and write*

$$z = \frac{w_3^2 + w_4^2}{w_1 w_2} \tag{12}$$

where $w_1 - w_4$ are jointly Gaussian with means and covariances identical to the means and covariances of $\xi_1 - \xi_4$. The probability density of (12) will be denoted by $p(z)$, and, as a consequence of the central limit theorem, we expect $p(z)$ to converge to the density of the true sample MSC as $N \to \infty$.

MOMENTS

In this section we shall obtain the means and covariances of the variables $w_1 - w_4$ and the mean and variance of z. In addition some aspects of the sampling procedure will be discussed.

Moments of $w_1 - w_4$

The means and covariances of $w_1 - w_4$ will be denoted by m_j and c_{jk} where $m_j = Ew_j = E\xi_j$ and $c_{jk} = E(w_j - m_j)(w_k - m_k) =$

---

[*] It will be convenient in the following to use z rather than z^2 in this definition even though z^2 would be more in keeping with the earlier definition of $\hat{\gamma}^2$.

$E(\xi_j - m_j)(\xi_k - m_k)$, $j,k = 1,\ldots,4$. The corresponding mean value column vector and covariance matrix will be denoted by $\underset{\sim}{m}$ and $\underset{\sim}{C}$. For the mean values we obtain $\underset{\sim}{m}^T = [\mu_1, \mu_2, \mu_3, \mu_4]$ where

$$\mu_1 = \mu_1(f_r) = \Delta \sum_{k=-(M-1)}^{(M-1)} \left(1 - \frac{|k|}{M}\right) R_{11}(k\Delta) \cos(2\pi f_r\, k\Delta) \tag{13}$$

$$\mu_2 = \mu_2(f_r) = \Delta \sum_{k=-(M-1)}^{(M-1)} \left(1 - \frac{|k|}{M}\right) R_{22}(k\Delta) \cos(2\pi f_r\, k\Delta) \tag{14}$$

$$\mu_3 = \mu_3(f_r) = \Delta \sum_{k=-(M-1)}^{(M-1)} \left(1 - \frac{|k|}{M}\right) R_{12}(k\Delta) \cos(2\pi f_r\, k\Delta) \tag{15}$$

$$\mu_4 = \mu_4(f_r) = \Delta \sum_{k=-(M-1)}^{(M-1)} \left(1 - \frac{|k|}{M}\right) R_{12}(k\Delta) \sin(2\pi f_r\, k\Delta)\,. \tag{16}$$

For the covariance matrix we write $\underset{\sim}{C} = \underset{\sim}{C}_o + \underset{\sim}{E}$ where

$$\underset{\sim}{C}_o = \frac{1}{\lambda}\begin{bmatrix} \mu_1^2 & (\mu_3^2+\mu_4^2) & \mu_1\mu_3 & \mu_1\mu_4 \\ (\mu_3^2+\mu_4^2) & \mu_2^2 & \mu_2\mu_3 & \mu_2\mu_4 \\ \mu_1\mu_3 & \mu_2\mu_3 & \frac{1}{2}(\mu_1\mu_2+\mu_3^2-\mu_4^2) & \mu_3\mu_4 \\ \mu_1\mu_4 & \mu_2\mu_4 & \mu_3\mu_4 & \frac{1}{2}(\mu_1\mu_2-\mu_3^2+\mu_4^2) \end{bmatrix} \tag{17}$$

$$\underset{\sim}{E} = \frac{1}{\lambda}\begin{bmatrix} \nu_1^2 & \nu_3^2 & \nu_1\nu_3 & 0 \\ \nu_3^2 & \nu_2^2 & \nu_2\nu_3 & 0 \\ \nu_1\nu_3 & \nu_2\nu_3 & \frac{1}{2}(\nu_1\nu_2+\nu_3^2) & 0 \\ 0 & 0 & 0 & -\frac{1}{2}(\nu_1\nu_2-\nu_3^2) \end{bmatrix} \tag{18}$$

with

$$\nu_1 = \nu_1(f_r) = \frac{\Delta}{M \sin(2\pi f_r \Delta)} \sum_{k=-(M-1)}^{(M-1)} R_{11}(k\Delta) \sin(2\pi f_r |k|\Delta)\,. \tag{19}$$

$$\nu_2 = \nu_2(f_r) = \frac{\Delta}{M \sin(2\pi f_r \Delta)} \sum_{k=-(M-1)}^{(M-1)} R_{22}(k\Delta) \sin(2\pi f_r |k| \Delta) \tag{20}$$

$$\nu_3 = \nu_3(f_r) = \frac{\Delta}{M \sin(2\pi f_r \Delta)} \sum_{k=-(M-1)}^{(M-1)} R_{12}(k\Delta) \sin(2\pi f_r |k| \Delta) \tag{21}$$

and, for $N \geq Q$,

$$\lambda = \lambda(N, Q) = \frac{6N^2 Q}{2N(2Q^2+1) - Q(Q^2-1)} \quad . \tag{22}$$

In practice M is usually chosen to be large and as a consequence it is often possible to neglect the matrix $\underset{\sim}{E}$ and approximate $\underset{\sim}{C}$ with $\underset{\sim}{C}_o$. The reason for this is that, as $M \to \infty$, the quantities $\mu_1 - \mu_4$ converge to the spectral densities of the sampled processes, while the quantities $\nu_1 - \nu_3$ converge to zero because of the factor M in the denominator of the defining expressions (19) - (21). That is, $\lim_{M\to\infty} \underset{\sim}{C} = \lim_{M\to\infty} \underset{\sim}{C}_o$.

An important quantity is the ratio ρ^2 defined by (9). This can be written as

$$\rho^2 = \rho^2(f_r) = \frac{m_3^2 + m_4^2}{m_1 m_2} = \frac{\mu_3^2 + \mu_4^2}{\mu_1 \mu_2} \tag{23}$$

and, under relatively mild assumptions, we will have

$$\lim_{\substack{M\to\infty \\ \Delta\to 0}} \rho^2(f_r) = \gamma^2(f_r) \quad . \tag{24}$$

This result plays an important role in the following.

Moments of z

Because the variables $w_1 - w_4$ are Gaussian, the moments of $z = (w_3^2 + w_4^2)/(w_1 w_2)$ are not defined. Consequently we cannot determine the moments of $\hat{\gamma}^2$ by simply equating them to the corresponding moments of z. However, we can obtain an asymptotic series for the moments of z that becomes an increasingly accurate approximation for the moments of $\hat{\gamma}^2$ as λ (equivalently N) becomes large.

For the mean and variance we have $Ez = \rho^2 + O(1/\lambda)$ and $\mathrm{Var}(z) = O(1/\lambda)$. As λ becomes large this implies that z converges in quadratic mean to ρ^2. For effective estimation of the MSC we wish ρ^2 to be as close to γ^2 as possible, and, as (24) demonstrates, this can be achieved with a negligibly small error by choosing M sufficiently large and Δ sufficiently small.

Another consequence of choosing M large is that the covariance matrix $\underset{\sim}{C}$ can be approximated by $\underset{\sim}{C}_o$ and this greatly simplifies the expressions for the moments. Using only the matrix $\underset{\sim}{C}_o$ and retaining terms up to $O(1/\lambda^2)$ we have for the mean and variance

$$Ez = \rho^2 + \frac{1}{\lambda}(1 - \rho^2)^2 + O(1/\lambda^2), \quad \underset{\sim}{C} = \underset{\sim}{C}_o \tag{25}$$

$$\mathrm{Var}(z) = \frac{2}{\lambda}\rho^2(1 - \rho^2)^2 + O(1/\lambda^2), \quad \underset{\sim}{C} = \underset{\sim}{C}_o \;. \tag{26}$$

DETERMINATION OF THE PROBABILITY DENSITY

Here we obtain an expression for the probability density of z and discuss some of its more important properties as they relate to overlap. It will be assumed throughout that the covariance matrix $\underset{\sim}{C}$ can be adequately approximated by the matrix $\underset{\sim}{C}_o$; the corresponding probability density of $w_1 - w_4$ will be denoted by $p_o(w_1, w_2, w_3, w_4)$. To distinguish between the density of z obtained using $\underset{\sim}{C}$ and the density obtained using $\underset{\sim}{C}_o$, the latter will be denoted by $p_o(z)$. It follows from the discussion above that $\lim_{M\to\infty} p(z) = \lim_{M\to\infty} p_o(z)$.

Determination of $p_o(z)$

We begin by writing $p_o(z)$ as the inverse Fourier transform of the associated characteristic function. After some elementary manipulation this appears as

$$p_o(z) = \frac{1}{2\pi}\int_{-\infty}^{\infty} dv \int_{-\infty}^{\infty} dw_1 \ldots \int_{-\infty}^{\infty} dw_4 |w_1 w_2|\, p_o(w_1, \ldots, w_4) \cdot \exp[-ivw_1w_2z + iv(w_3^2 + w_4^2)] \;. \tag{27}$$

This integral is difficult to evaluate because of the absolute value term $|w_1w_2|$ in the integrand. However, it can be simplified by observing that when $z > 0$ there is no contribution to the integral from the region defined by $w_1w_2 < 0$. Since we are only interested in positive values of z we may replace $|w_1w_2|$ by (w_1w_2) in (27) whenever $z > 0$. The evaluation of (27) is a lengthy procedure and space limitations do not permit us to present the details here. We simply state the final result which appears as

$$p_o(z) = \frac{1}{2\pi i}\int_{-i\infty}^{i\infty} D^{-1/2}(v)\{\lambda P^2(v) + R(v)\}\exp[-\lambda E(v)]dv, \quad z > 0 \tag{28}$$

where P(v), R(v), D(v) and E(v) are given by

$$P(v) = \frac{(z - v)}{(1 - \rho^2)^{1/2}(v_o + v)(v_1 - v)} \tag{29}$$

$$R(v) = \frac{z\rho^2 + [\rho^2 - z(1 - \rho^2)]v + (1 - \rho^2)v^2}{(1 - \rho^2)(1 - v)(v_o + v)(v_1 - v)} \tag{30}$$

$$D(v) = (z - v)(1 - v)(v_1 + v)(v_2 - v) \tag{31}$$

$$E(v) = \frac{v\left(\frac{z - \rho^2}{1 - \rho^2} - v\right)}{(v_o + v)(v_1 - v)} \tag{32}$$

$$v_\ell = \frac{1}{2(1 - \rho^2)}\left\{(-1)^\ell (1 + \rho^2)(1 - z) + \sqrt{(1+\rho^2)^2(1-z)^2 + 4z(1-\rho^2)^2}\right\}, \quad \ell = 0, 1 \quad . \tag{33}$$

Further evaluation of (28) is very difficult, but as we shall see this evaluation will not be necessary.

The essential feature of the above expression for $p_o(z)$ is that the functions $P(v)$, $R(v)$, $D(v)$ and $E(v)$ depend only on z and ρ^2, and the overlap enters only via the parameter λ, which depends on N and Q. Clearly all combinations of N and Q that result in the same value for λ, yield identical expressions for $p_o(z)$, and, as a consequence, we can compensate for overlap by appropriately modifying N.

Effective Number of Degrees of Freedom

If there is no overlap $(Q = 1)$, the sample MSC is obtained by averaging N independent transforms formed from the original processes. In this case we say that the problem has N degrees of freedom.

Suppose now that there is overlap present. In this case define $N_e = \lambda(N, Q)$ with $\lambda(N, Q)$ given by (22). Obviously the function $p_o(z)$ obtained in the overlap case with N and Q is identical to the $p_o(z)$ obtained with no overlap and N_e degrees of freedom (the value for λ is the same in both cases). We shall refer to N_e as the effective number of degrees of freedom for the problem. When $N > 1$ it can be shown that $N_e \le N$ with equality only when there is no overlap.

Reduction of the Bias and Variance

An important reason for using overlap processing is that it reduces the bias and variance of the sample MSC without requiring additional input data. This reduction is accomplished by increasing the effective value of λ. The benefits of overlap are limited, however, as the following argument demonstrates.

Consider first the no-overlap case in which N discrete Fourier transforms are used to form the sample MSC. In this case, $\lambda = N$ and we further assume that $N > 1$. Now suppose overlap processing is employed using the same data. The overlap is characterized by the integer Q, and the number of discrete Fourier transforms now available* is $N' = (N-1)Q+1$. Because $N > 1$ we have $N' \ge Q$ and from (22) we obtain

$$\lambda' = \lambda \frac{6Q(NQ - Q + 1)^2}{2N(NQ - Q + 1)(2Q^2 + 1) - NQ(Q^2 - 1)} \quad . \tag{34}$$

---

*These transforms are assumed to be formed from observations having the same length as in the no-overlap case, viz. K.

It is readily verified that $\lambda' \leq 3/2\ \lambda$ with equality attained only in the limit as $N \rightarrow \infty$ and $Q \rightarrow \infty$.

As a consequence of the above it is clear that the effective value of λ cannot be increased more than 50% through the use of overlap. If $\rho^2 \approx \gamma^2$, it follows from (25) and (26) that both the bias and variance have leading terms that are proportional to $1/\lambda$.* Therefore, regardless of the amount of overlap, the bias and variance cannot be reduced to less than 2/3 of their no-overlap values** (neglecting higher order terms in $1/\lambda$).

Accuracy of the Gaussian Approximation

The fundamental assumption that enabled us to derive the expression (28) for $p_0(z)$ was that the joint density of $\xi_1 - \xi_4$ can be approximated by a Gaussian density. This assumption was based on the central limit theorem and, as often happens in limit theorem applications, precise bounds on the error are not available. To obtain an indication of the validity of the Gaussian approximation, the integral (28) has been evaluated numerically in several cases and compared to the exact density.

Since overlap can be accounted for by appropriately adjusting the number of degrees of freedom, it is clearly sufficient to consider only the no-overlap case ($Q = 1$, $\lambda = N$). For the exact density we use Goodman's result (8) with $\hat{\gamma}^2 = z$. The expressions (8) and (28) have been evaluated for $\rho^2 = 0, 0.25, 0.50, 0.75$ and for $N = 50, 100$. The results are presented in Figures 1 and 2. As expected the agreement is best in the vicinity of $z = \rho^2$ and improves as N increases.

REFERENCES

[1] G. C. Carter, "Time Delay Estimation," Naval Underwater Systems Center Technical Report No. 5335, April 1976.

[2] G. C. Carter, "Estimation of the Magnitude-Squared Coherence Function (Spectrum)," Naval Underwater Systems Center Technical Report No. 4343, May 1972.

[3] D. Becker, "Investigate the Use of Alternate Sample Statistics," ORINCON Company Report San Diego, CA, 1976.

[4] D. Amos and L. H. Koopmans, "Tables of the Distribution of the Coefficient of Coherence for Stationary Bivariate Gaussian Processes," Sandia Corporation Monograph SCR-483, March 1963.

[5] N. R. Goodman, "On the Joint Estimation of the Spectra, Cospectrum and Quadrature Spectrum of a Two-Dimensional Stationary Gaussian Process," Scientific Paper No. 10, Engineering Statistics Laboratory, New York University, March 1957.

[6] G. C. Carter, C. H. Knapp and A. H. Nuttall, "Estimation of the Magnitude-Squared Coherence Function via Overlapped Fast Fourier Transform Processing," IEEE Trans. on Audio and Electroacoustics, Vol. AU-21, pp. 337-344, 1973.

[7] A. Greenstein, "Inter Sensor Coherent Processing," General Electric Company, Heavy Military Electronics Division, Report DA 140-525, January 1976.

[8] H. Cramér, Mathematical Methods of Statistics, Princeton University Press, Princeton, New Jersey, 1946.

[9] S. Bochner, Lectures on Fourier Integrals, Princeton University Press, Princeton, New Jersey, 1959.

---

*An exception occurs in the special case where $\rho^2 = 0$; here the leading term of the variance is proportional to $1/\lambda^2$.

**Except that the reduction for the variance is 4/9 when $\rho^2 = 0$.

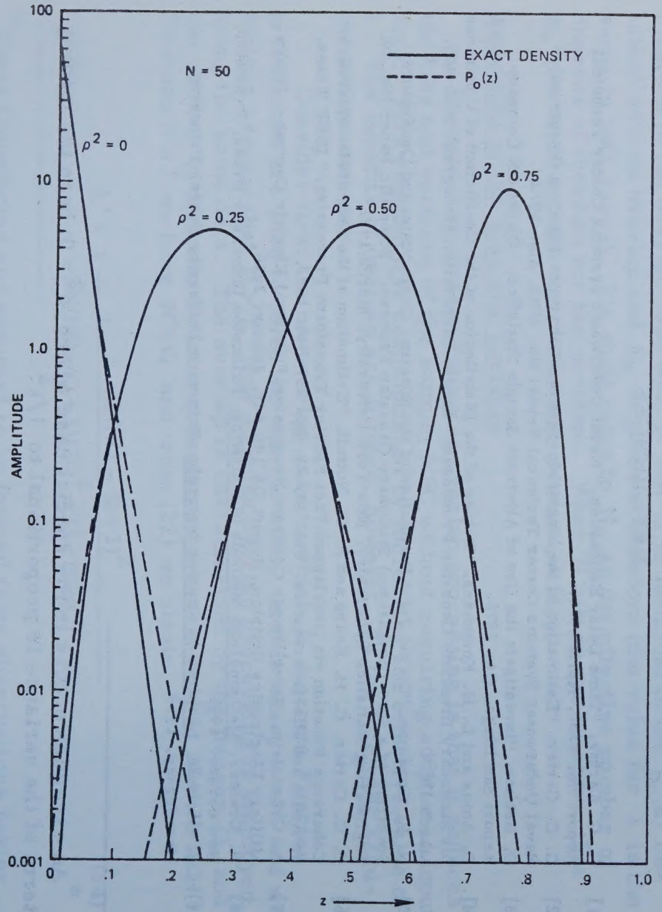


Figure 1. Comparison of $P_o(z)$ and the exact density, N = 50.

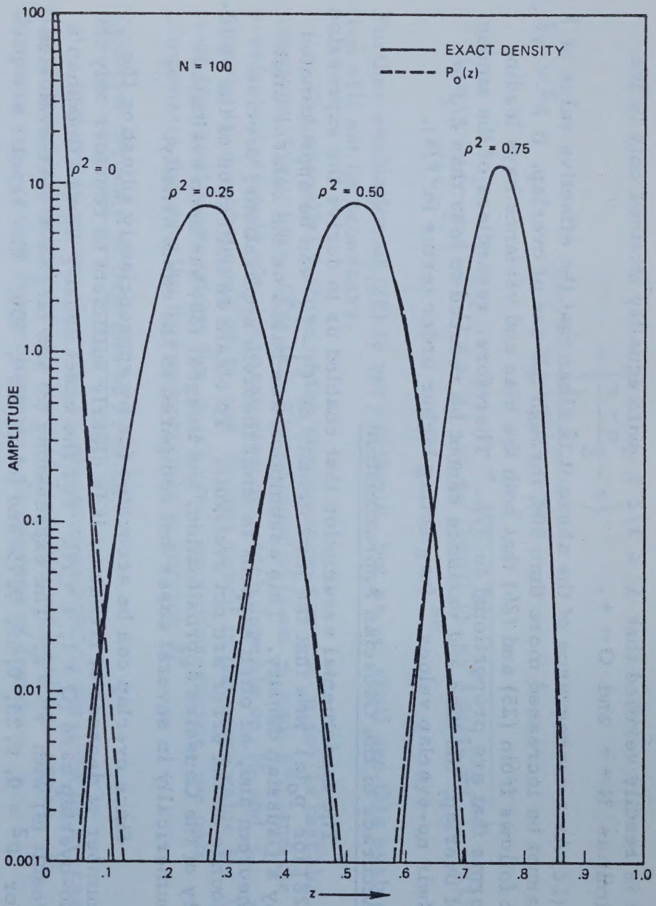


Figure 2. Comparison of $P_o(z)$ and the exact density, N = 100.

EVALUATION OF ERROR PROBABILITIES FOR MPSK SATELLITE COMMUNICATION SYSTEMS WITH INTERFERENCE

J. K. OMURA
University of California
Los Angeles, California

T. C. HUANG
LinCom Corporation
Pasadena, California

P. C. JAIN
Defense Communication Agency
Arlington, Virginia

ABSTRACT

This paper presents a method of evaluating error probabilities of coherent MPSK modulation systems with conventional nonlinear satellite repeaters when there is uplink noise, downlink noise, a phase reference error, and interference. The numerical evaluations are based on a generalization of the classical moment technique for approximating probability distributions.

INTRODUCTION

Figure 1 shows a diagram of the coherent MPSK satellite communication system which we examine in this paper. Here we have both uplink and downlink noise as well as a general interference signal on the uplink represented by i(t). We assume a conventional satellite repeater with both AM/AM and AM/PM nonlinearities. Conventional coherent MPSK modulators and demodulator models are assumed in this work as well as receiver phase reference errors.

We begin by describing the mathematical models for the transmitter, satellite repeater, and receiver for this coherent MPSK satellite communication system. A mathematical expression for the uncoded bit error probability is derived next followed by a discussion of the "moment technique" used to evaluate the error probabilities for various system parameters. Numerical results are shown at the end.

This paper is based on work conducted at Lin Com Corporation under a contract from the Defense Communication Agency. The final report for this contract [1] contains more details of this paper as well as noncoherent MFSK systems, jamming interference and anti-jam analysis, processing satellites, and a queueing analysis for multiple users. Reference to earlier works in this area are also cited there.

SYSTEM MODEL

For the single user and receiver terminal pair, we have the model of Figure 1. This model also applies to multiple users and receivers in a TDMA mode of operation.

Transmitter

We consider MPSK signals given by

$$m(t) = A\cos[\omega_c t+\phi],\ 0 \le t \le T$$

where T = Baud Time (MPSK Signal Duration) and the symbol transmitted is the phase $\phi \epsilon \left\{ \frac{2\pi}{M} m: \; m = 0,1,\ldots M-1 \right\}$. We restrict ourselves to the conventional number of phases given by $M = 2^k$, $k = 1,2,3,4$, so that k bits are mapped into one of $M = 2^k$ phases during each baud time. For the uncoded case this mapping from k bits to one of $M = 2^k$ phases is done with Gray Codes. (This is not an error correcting code.) It has the property that if the demodulator incorrectly chooses a phase adjacent to the transmitted phase then only one of the $k = \log_2 M$ bits would be demodulated incorrectly. Since such adjacent phase errors are the dominant type of symbol errors we have the approximation that the uncoded bit error probability P_b is given by

$$P_b = \frac{P_e}{\log_2 M}$$

where P_e is the probability of a phase symbol demodulation error.

When there is no distortion of the modulated signal due to the transmit filter then the transmitted signal $0 \leq t \leq T$ has the form,

$$s(t) = m(t)$$

$$= \sqrt{\mathscr{E}_s} \cos\phi \; \phi_c(t) + \sqrt{\mathscr{E}_s} \sin\phi \; \phi_s(t)$$

where we define symbol energy

$$\mathscr{E}_s \triangleq A^2 T/2$$

and the normalized cosine and sine coordinates

$$\phi_c(t) = \sqrt{2/T} \cos \omega_c t \;, \quad \phi_s(t) = -\sqrt{2/T} \sin \omega_c t$$

for $0 \leq t \leq T$. Here

$$\mathscr{E}_s = \int_0^T s^2(t) dt$$

when $\omega_c T$ is a multiple of π or if $\omega_c T >> \pi$ which we can always assume for the frequencies of interest.

Note that for fixed amplitude A and baud time T, the energy per symbol, $\mathscr{E}_s$, is the same regardless of the choice of $M = 2^k$. The energy per uncoded bit, $\mathscr{E}_b$, is different, however, and given by

$$\mathscr{E}_b = \frac{\mathscr{E}_s}{\log_2 M}$$

When there is adequate power

$$P = A^2/2$$

and the channel is bandlimited, then T is typically chosen as small as possible and still meet the bandwidth restriction.* Hence it is natural to consider fixed P and T for all M = 2,4,8, and 16. This means that in comparing the performance for various M = 2,4,8, and 16 we should use the same signal energy $\mathscr{E}_s$. Traditionally when we were power limited there were no bandwidth limitations and T was unrestricted. There the common parameter chosen was $\mathscr{E}_b$ for comparing the performance of various choices of M. Since we are primarily interested in throughput over bandlimited channels for a fixed bit error probability, we use signal energy ε_s as the common parameter in comparing different modulations.

Uplink Noise, Downlink Noise, and Interference

The uplink noise process $\tilde{n}_1(t)$ and the downlink noise process $\tilde{n}_2(t)$ are assumed to be zero mean white Gaussian noise processes with autocorrelation

$$E\{\tilde{n}_2(t+\tau)\tilde{n}_2(t)\} = \frac{N_0^{(1)}}{2}\,\delta(\tau)$$

$$E\{\tilde{n}_2(t+\tau)\tilde{n}_2(t)\} = \frac{N_0^{(2)}}{2}\,\delta(\tau)$$

where $N_0^{(1)}$ and $N_0^{(2)}$ are the single-sided noise spectral densities.

The uplink noise must pass through the satellite front end bandpass filter. If we assume that the bandpass filter essentially passes the transmitted MPSK signal undistorted and filters out all inputs outside the signal space then the bandpass filtered uplink noise can be represented in terms of the signal sine and cosine coordinates

$$n_1(t) = n_{1c}\phi_c(t) + n_{1s}\phi_s(t)$$

where n_{1c} and n_{1s} are independent zero mean Gaussian random variables with variance $N_0^{(1)}/2$. The uplink signal-to-noise ratio is defined as the total signal energy ε_s divided by the variance of all the noise components in the signal space. Thus the uplink signal-to-noise ratio is given by

$$\rho_1^2 \triangleq \frac{\mathscr{E}_s}{\mathrm{Var}[n_{1c}+n_{1s}]} = \frac{A^2T}{2N_0^{(1)}}$$

The downlink noise enters the receiver front and filter which also is assumed to pass an ideal MPSK signal undistorted. Since the receiver uses sine and cosine coordinates with phase reference denoted $\bar{\phi}$ we have the filtered downlink noise represented as

$$n_2(t) = n_{2c}\hat{\phi}_c(t) + n_{2s}\hat{\phi}_s(t)$$

---

*Later we shall examine the problem of intersymbol interference created when T is reduced further.

where

$$\hat{\phi}_c(t) = \sqrt{\frac{2}{T}}\cos[\omega_c t+\phi] \;, \quad \hat{\phi}_s(t) = -\sqrt{\frac{2}{T}}\sin[\omega_c t+\bar{\phi}]$$

and n_{2c} and n_{2s} are zero mean Gaussian random variables with variance $N_0^{(2)}/2$. If the downlink signal envelope at the receiver has a peak or maximum possible value of f_0 then the downlink signal-to-noise ratio[*] is defined to be

$$\rho_2^2 = \frac{f_0^2 T}{2N_0^{(2)}}$$

This peak power satellite output is used to define downlink signal-to-noise ratio regardless of the true satellite output power. For the ideal hard limiter satellite this is the true signal-to-noise ratio.

For both uplink and downlink noise, we consider the filtered noise as the projection of the unfiltered white Gaussian noise process onto the signal space. This representation is useful for analysis and can be easily generalized to other forms of modulation signals. Also we feel it is a good model for the uplink and downlink noise processes in satellite communication systems.

The interference signal i(t) is assumed to lie in the signal space and have the form

$$i(t) = I_c\phi_c(t) \pm I_s\phi_s(t)$$

where I_c and I_s are the projections of the interference in the signal coordinates ϕ_c and ϕ_s.

Satellite Repeater Model

Basic to modeling a satellite repeater is the notion of a memoryless nonlinearity followed by a zonal filter commonly referred to as a bandpass nonlinearity (BPNL). A BPNL is characterized by an AM/AM function and an AM/PM function. The AM/AM function describes how the amplitude of the BPNL narrowband input signal is converted to the output amplitude while the AM/PM function describes how the output phase shift is related to the input envelope.

A soft limiter followed by a zonal filter has a general AM/AM function $f_1(\cdot)$ and an AM/PM function $g_1(\cdot) \equiv 0$. That is, there is no phase distortion with limiters. A TWT followed by a zonal filter has general AM/AM function $f_2(\cdot)$ and AM/PM function $g_2(\cdot)$. These functions can be obtained directly from measurements and are often data supplied by the manufacturer of the TWT. When we cascade two such BPNLs we obtain another BPNL with AM/AM function

$$f(\cdot) = f_2[f_1(\cdot)]$$

---

[*] For the ideal linear satellite repeater we use average energy in place of $f_0^2T/2$.

and AM/PM function

$$g(\cdot) = g_2[f_1(\cdot)] + g_1(\cdot)$$

A narrowband signal of the form

$$x(t) = \nu(t)\cos[\omega_c t+\eta(t)]$$

into a BPNL has then the output

$$z(t) = f(\nu(t))\cos[\omega_c t+\eta(t) + g(\nu(t))]$$

Special cases of our general satellite repeater model include:

● Linear	● Hard Limiter
$f(\nu(t)) = G\nu(t)$	$f(\nu(t)) = G$
$g(\nu(t)) = 0$	$g(\nu(t)) = 0$

where G is the gain of the satellite repeater. In this paper we emphasize these two special forms since they are easy to parameterize and represent the two extreme forms of satellite repeaters.

As shown in Figure 1 the satellite repeater is modeled as a cascade of a bandpass filter followed by a general BPNL which actually is a cascade of a soft limiter and TWT. With no intersymbol interference at the transmitter the signal at the input to the bandpass filter consists of signal plus white Gaussian noise plus interference. Assuming that the bandpass filter passes the signal and interference undistorted and passes only that part of the uplink white Gaussian noise that projects onto the signal space, we model the bandpass filter output as

$$x(t) = s(t) + n_1(t) + i(t)$$

$$=\sqrt{\frac{T}{2}}\,\nu\cos[\omega_c t+\eta]$$

where

$$\nu = \sqrt{(\sqrt{\mathscr{E}_s}\cos\phi+n_{1c}+I_c)^2 + (\sqrt{\mathscr{E}_s}\sin\phi+n_{1s}+I_s)^2}$$

and

$$\eta = \tan^{-1}\left(\frac{\sqrt{\mathscr{E}_s}\sin\phi + n_{1s}+I_s}{\sqrt{\mathscr{E}_s}\cos\phi + n_{1c}+I_c}\right)$$

Note that the envelope ν and phase η are random variables that are constant during the baud interval. The satellite repeater output is then given by

$$z(t) = f(\nu)\cos[\omega_c t+\eta+g(\nu)] \quad 0 \le t \le T$$

where again $f(\nu)$ and $g(\nu)$ are random variables.

Receiver

For the general case of interest here, there is uplink noise which passes through the satellite repeater causing the satellite repeater to transmit some stochastic signal which is no longer a Gaussian process. For this general case there are no known optimum receivers. In practice the demodulator following the receiver filter usually consists of an integrate and dump operation on each of the quadrature samples are used to make a decision about the transmitted MPSK signal's phase. The integrate and dump operation is sometimes approximated by a low pass filter whose output is sampled. This conventional receiver is shown in Figure 1.

Assuming the receiver filter leaves the satellite transmitted signal undistorted, the received signal out of the receiver filter is assumed to have the form

$$r(t) = z(t) + n_2(t) = r\hat{\phi}_c + \tilde{r}\hat{\phi}_s$$

where

$$r = z + n_{2c}, \quad \tilde{r} = \tilde{z} + n_{2s}$$

and

$$z = \sqrt{T/2}\; f(\nu)\cos[\eta+g(\nu)-\overline{\phi}], \quad \tilde{z} = \sqrt{T/2}\; f(\nu)\sin[\eta+g(\nu)-\overline{\phi}]$$

The conventional detector obtains the sine and cosine pair $(r,\tilde{r})$ and bases its decision as to which one of M phases was transmitted according to the phase cone in which $(r,\tilde{r})$ lies. Figure 2 illustrates the decision regions for $M = 2,4,$ and 8. For fixed uplink noise the downlink signal pair $(z,\tilde{z})$ is then shifted by the downlink noise pair $(n_{2c},\tilde{n}_{2s})$ resulting in the detector output pair $(r,\tilde{r})$ upon which the decision is based.

UNCODED BIT ERROR PROBABILITY

Without loss of generality for MPSK modulation, assume that during the baud time $[0,T]$ the transmitted phase is $\phi = 0$. The other possible phases are $(2\pi/M)m$ for $m = 1,2,\ldots,M-1$. The receiver bases its decision on the pair $(r,\tilde{r})$ where an error occurs if and only if $(r,\tilde{r})$ falls outside the cone of $2\pi/M$ radians centered at $\theta = 0°$. This decision region is shown in Figure 2 where it is denoted R_0.

To compute the uncoded error probability we first condition on assuming the satellite output

$$z(t) = f(\nu)\cos[\omega_c t+\eta+g(\nu)] \quad 0 \leq t \leq T$$

is known. This means the vector $(z,\tilde{z})$ is known when we also assume the phase reference $\overline{\phi}$ is fixed. Then the conditional error probability is

$$P_e(z,\tilde{z};\overline{\phi}) = \Pr\{(r,\tilde{r})\notin R_0 \,|\, z,\tilde{z};\overline{\phi}\}$$

For M = 2 we have the form

$$P_e(z,\tilde{z};\overline{\phi}) = Q\left(\sqrt{\frac{2}{N_0^{(2)}}}\; z\right) = Q\left(\sqrt{2\rho_2^2}\,\hat{f}(\nu)\cos[\eta+g(\nu)-\overline{\phi}]\right)$$

where

$$\hat{f}(\nu) = f(\nu)/f_0$$

and

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

For $M = 2^k$, $k = 2,3,\ldots$, we have the bound (see Figure 2)

$$P_e(z,\tilde{z};\overline{\phi}) \leq Q\left(\sqrt{\frac{2}{N_0^{(2)}}}\, d_1(z,\tilde{z})\right) + Q\left(\sqrt{\frac{2}{N_0^{(2)}}}\, d_2(z,\tilde{z})\right)$$

where

$$d_1(z,\tilde{z}) = z \sin\frac{\pi}{M} - \tilde{z}\cos\frac{\pi}{M} = \sqrt{\frac{T}{2}}\, f(\nu)\sin\left[\frac{\pi}{M} - \eta - g(\nu) + \overline{\phi}\right]$$

$$d_2(z,\tilde{z}) = z \cos\frac{\pi}{M} - \tilde{z}\sin\frac{\pi}{M} = \sqrt{\frac{T}{2}}\, f(\nu)\cos\left[\frac{\pi}{M} + \eta - g(\nu) - \overline{\phi}\right]$$

and the envelope ν and phase η are related to the uplink noise and interference components. Defining

$$F(n_{1c}+I_c, n_{1s}+I_s;\overline{\phi}) = \begin{cases} Q\left(\sqrt{2\rho_2^2}\,\hat{f}(\nu)\cos[\eta+g(\nu)-\overline{\phi}]\right), & M = 2 \\ Q\left(\sqrt{2\rho_2^2}\,\hat{f}(\nu)\sin[\frac{\pi}{M} - \eta - g(\nu) + \overline{\phi}]\right) \\ \quad + Q\left(\sqrt{2\rho_2^2}\,\hat{f}(\nu)\cos[\frac{\pi}{M} + \eta + g(\nu) - \overline{\phi}]\right), & M > 2 \end{cases}$$

where ν and η depend on the uplink noise plus interference components, $n_{1c} + I_c$ and $n_{1s} + I_s$, we see that the unconditioned error probability for given phase reference $\overline{\phi}$ is

$$P_e(\overline{\phi}) = E[P_e(z,\tilde{z};\overline{\phi})]$$

$$\leq E\{F(n_{1c}+I_c, n_{1s}+I_s;\overline{\phi})\}$$

Here the expectation is over the noise components n_{1c} and n_{2c} which are independent zero mean Gaussian random variables and the interference terms I_c and I_s.

The evaluation of the error probability $P_e(\overline{\phi})$ reduces to the evaluation of the above expectation. Next we present a technique for computing this expectation which is a generalization of the classical moment approximation for probability distributions.

NUMERICAL EVALUATION OF ERROR PROBABILITIES

When there is no uplink interference we had learned that, from a numerical computation point of view, direct integrations using Gauss Quadrature rules [2] were usually more efficient and general for error probabilities than taking various series expansions of the integrand and obtaining a sum of simpler terms. This motivated us to investigate the generalized form of the Gauss-Quadrature rules referred to as the classical "Hamburger Moment Problem." The Russian mathematicians Chebyshev, Markov and Krein [3] expanded this theory and recently it was applied to simple communication systems by Benedetto and Biglieri [4] and Yao and Biglieri [5]. We have successfully tested with computer algorithms, a new generalization of this moment technique which allows us to numerically evaluate the performance of the more complex satellite communication systems of this paper.

EXAMPLES

Numerical evaluation of uncoded error probabilities are shown for two examples in Figures 3 and 4. With some modification to allow for signal distortions, we compute the effect of Chebychev filters on QPSK signals over a hard limiter satellite. Here we assume perfect receiver phase reference and various values of the band time and 3 dB filter bandwidth product, BT. Figure 4 shows the effect of receiver reference phase error $\psi = \overline{\Phi} - \phi_0$ where ϕ_0 is the ideal reference phase. Here we also have a Butterworth filter with BT = 1.0. M = 2,4, and 8 are compared for both the ideal linear and hard limiter satellite repeaters. Even for M = 8, the effect of up to 20 intersymbol interference terms can be easily computed.

Our experience with computing many error probabilities of these kind shows that the "moment" technique is both accurate and fast. The computation only grows linearly with the number of independent random variables that sum to form the interference.

REFERENCES

[1] Lindsey, W. C., Omura, J. K., Huang, T. C., and Biederman, L., "Throughput Study for Military Satellite Communications Under Jamming," Final Report on Contract DCA 100-77-C-0052, Volumes I, II, III, IV, and V, September 1978.

[2] Krylov, V. I., Approximate Calculation of Integrals, translated by A. H. Stroud, MacMillian, New York, 1962.

[3] Krein, M. G., "The Ideas of P. L. Cebyshev and A. A. Markov in the Theory of Limiting Values of Integrals and Their Further Developments," American Mathematical Soc. Transl. Series 2, Vol. 12, 1951.

[4] Benedetto, S., Biglieri, E., and Castellani, V., "Combined Effects of Intersymbol, Interchannel, and Co-channel Interference in M-ary CPSK Systems," IEEE Transactions on Communications, Vol. COM-21, pp. 997-1008, September 1973.

[5] Yao, K., and Biglieri, E., "Moment Inequalities for Error Probabilities in Digital Communications," Proceedings of the National Telecommunications Conference, December 1977.

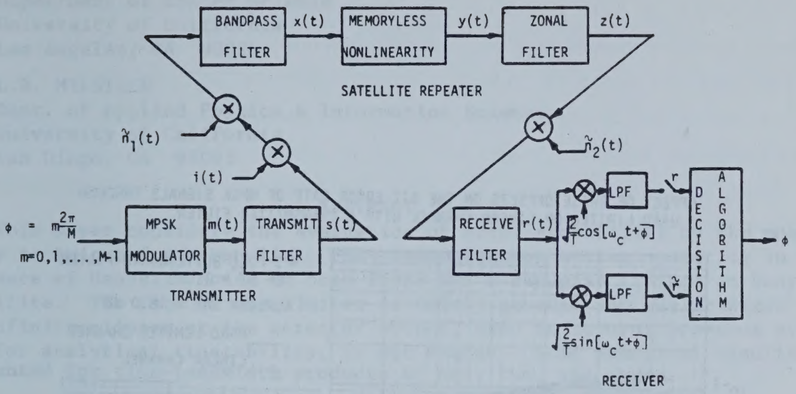
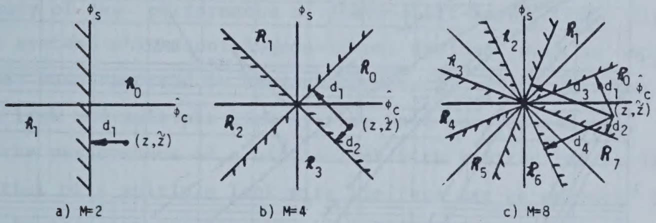

BANDPASS FILTER
x(t)
MEMORYLESS NONLINEARITY
y(t)
ZONAL FILTER
z(t)
SATELLITE REPEATER
$\tilde{n}_1(t)$
i(t)
$\tilde{n}_2(t)$
$\phi = m\frac{2\pi}{M}$
m=0,1,...,M-1
MPSK MODULATOR
m(t)
TRANSMIT FILTER
s(t)
TRANSMITTER
RECEIVE FILTER
r(t)
$\sqrt{\frac{2}{T}}\cos[\omega_c t+\hat{\phi}]$
$\sqrt{\frac{2}{T}}\sin[\omega_c t+\hat{\phi}]$
LPF
LPF
r
$\tilde{r}$
DECISION
ALGORITHM
$\hat{\phi}$
RECEIVER

Figure 1. MODEL FOR SINGLE CHANNEL MPSK SYSTEM.


ϕ_s
$\hat{\phi}_c$
R_0
R_1
d_1
$(z,\tilde{z})$
R_2
R_3
d_2
d_3
d_4
R_4
R_5
R_6
R_7

a) M=2 b) M=4 c) M=8

Figure 2. MPSK MEMORYLESS DECISION REGIONS.

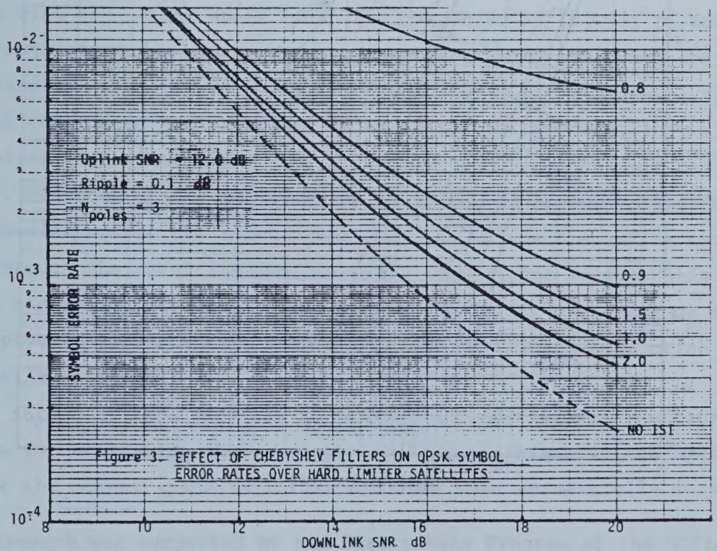
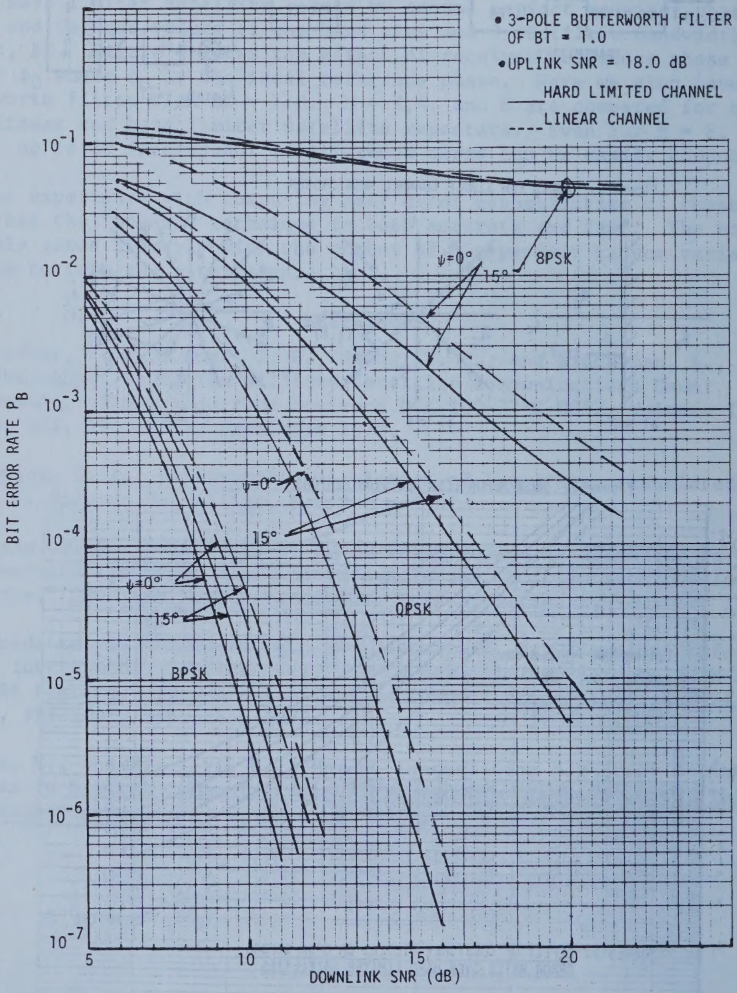


Figure 3. EFFECT OF CHEBYSHEV FILTERS ON QPSK SYMBOL ERROR RATES OVER HARD LIMITER SATELLITES

EFFECT OF PHASE OFFSETS ON THE BIT ERROR RATE OF MPSK SIGNALS THROUGH A HARD LIMITED OR LINEAR CHANNEL WITH A TRANSMITTER FILTER



PERFORMANCE OF PSK SYSTEMS WITH BANDPASS LIMITERS

K. YAO *
Department of System Science
University of California
Los Angeles, CA 90024

L.B. MILSTEIN
Dept. of Applied Physics & Information Science
University of California
San Diego, CA 92093

ABSTRACT

This paper considers the evaluation of error probability by the moment space technique for a two-link PSK communication system operating in the presence of Gaussian noise on both links and a bandpass limiter on-board the satellite. The form of the limiter is fairly general and the presence of an infinite clipper at the detector output, used in several previous studies for analytical tractability, is not needed. Some numerical results are presented for time-bandwidth products of one, two, and three.

I. INTRODUCTION

The study of the performance of phase-shift-keyed (PSK) digital communnication systems which contain non-linear devices, such as hard and soft limiters, has been reported by various authors [1]-[5]. In these studies, both single link and multiple link systems have been considered. It is known that the performance of a single link with limiter can be degraded [2], while that of a multiple link with limiters can be enhanced [3]-[5], as compared to the corresponding linear system.

In this paper, the model described in [2] and [3] will be used in conjunction with the moment space technique presented in [6]-[8] to bound the error probability of a two-link frequency-translating satellite system with an on-board non-linear device which is typically a traveling-wave tube (TWT). The model considered here consists of an uplink to the satellite, a bandpass limiter on the satellite, and a downlink to a coherent matched filter detector. In both uplink and downlink channels, independent bandlimited white Gaussian noises are added to the signal. Since the uplink noise is passed through the satellite nonlinearity, this output is generally a non-Gaussian stochastic process. Then, matched filtering operation on this stochastic process yields a test statistic whose distribution function is not analytically tractable.

The intent of this paper is to provide a computationally efficient technique for evaluating the error probability of this two-link system. We note that the moment space technique has been used successfully in evaluat-

---

*This research was supported by the Electronics Program of the Office of Naval Research.

ing the error probabilities of intersymbol interference, co-channel interference, and spread-spectrum multiple-access interference problems (see [6]-[8]). By applying this technique to the two-link satellite problem, one can use a model of limiter more general than the simple hard limiter. Furthermore, one can avoid the somewhat artificial use of an infinite clipper (i.e., a device which implements the signum function) at the detector output for implementing majority logic decision in order to compute error probability. Thus, the present approach allows the evaluation of performance of a more realistic overall system.

II. MODEL OF THE SYSTEM

The model of the two-link satellite system is shown in Fig. 1.

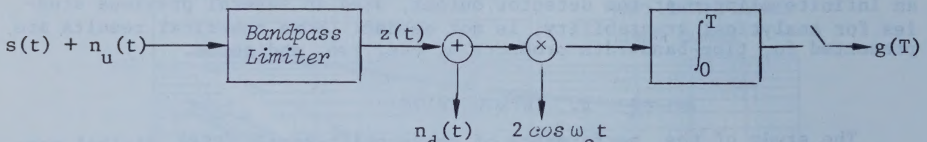


FIG. 1 - Model of a two-link satellite system.

The binary PSK signal $s(t)$ is given by

$$s(t) = \begin{cases} \pm A \cos\omega_o t\ , & 0 \le t \le T \\ 0\ , & \text{elsewhere}, \end{cases}$$

where $\pm A$ occurs with equal probability and $n_u(t)$ and $n_d(t)$ are the uplink and downlink noise processes, respectively. Both are assumed to be wide-sense stationary, zero-mean, narrow-band Gaussian random processes independent of each other and each with its own flat power spectral density in the frequency band $\omega_o - \omega_c \le |\omega| \le \omega_o + \omega_c$. The bandpass limiter is modeled as an instantaneous nonlinearity followed by a zonal filter centered around the carrier frequency ω_o. The instantaneous nonlinearity is shown in Fig.2 .

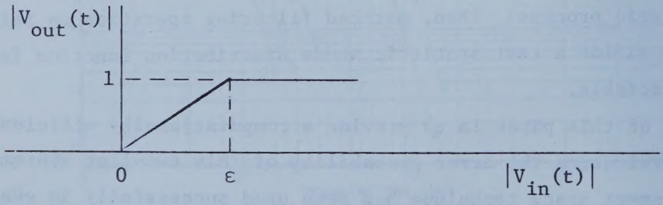


FIG. 2 - Instantaneous nonlinearity.

We note that, for ε equal to zero, we obtain a hard limiter while, as ε increases from zero, we obtain various forms of soft limiter.

Finally, the signal is detected by the use of a coherent receiver. That is, the receiver performs a coherent demodulation followed by a matched filter detection. If $z(t)$ is the bandpass limiter output, then

$$z(t) = V_{out}(t) \cos (\omega_o t + \theta(t)) ,$$

and $g(T)$, the final statistic, is given by

$$g(T) = \int_0^T [V_{out}(t) \cos \theta(t) + x_d(t)]dt , \tag{2.1}$$

where

$$n_u(t) = x_u(t) \cos \omega_o t - y_u(t) \sin \omega_o t \quad ,$$
$$n_d(t) = x_d(t) \cos \omega_o t - y_d(t) \sin \omega_o t \quad ,$$
$$\theta(t) = \tan^{-1}[y_u(t)/(A+x_u(t))] \quad .$$

Clearly, $x_u(t)$, $y_u(t)$, $x_d(t)$, and $y_d(t)$ are all independent baseband Gaussian random processes bandlimited to $|\omega| \leq \omega_c$. Upon defining

$$g_1(T) = \int_0^T V_{out}(t) \cos \theta(t)dt , \tag{2.2a}$$

$$g_2(T) = \int_0^T x_d(t)dt \quad , \tag{2.2b}$$

we can rewrite Eq.(2.1) as

$$g(T) = g_1(T) + g_2(T) \quad . \tag{2.3}$$

While $g_2(T)$ is a zero-mean Gaussian random variable, $g_1(T)$, being the integral of a non-Gaussian random process, has an analytically intractable distribution function. In order to overcome this basic problem, we replace Eq.(2.2a) by the following approximation,

$$g_1(T) \simeq \sum_{i=1}^{N} V_{out}(iT/N) \cos \theta(iT/N) , \tag{2.4}$$

where $N = 2f_cT$, $f_c = \omega_c/2\pi$, and the product f_cT is assumed to be a positive integer. This approximation permits analytical tractability, since now $g_1(T)$, given by Eq.(2.4), is the sum of a finite number of independent random variables. This independence follows from the Nyquist sampling of the bandlimited white Gaussian process $n_u(t)$. In particular, let

$$\alpha_i = V_{out}(i/2f_c)\cos\,\theta(i/2f_c) \qquad (2.5)$$

Then, in [2] , it is shown that the probability density of any α_i is given by

$$f_\alpha(\alpha) = \begin{cases} 0 & , \; |\alpha| > 1 \;, \\ 2\varepsilon'\phi(\varepsilon'\alpha - \lambda)\{\Phi[(1-\alpha^2)^{1/2}\,\varepsilon'] - \frac{1}{2}\} & \\ +[2/\pi(1-\alpha^2)]^{1/2}\,exp[-\lambda^2(1-\alpha^2)/2] \;, & |\alpha| \le 1 \;, \\ \cdot[\phi(\lambda\alpha - \varepsilon') + \lambda\alpha\,\Phi(\lambda\alpha - \varepsilon')] & \end{cases} \qquad (2.6)$$

where $\lambda = A/\sigma_1$, $\varepsilon' = \varepsilon/\sigma_1$, $\sigma_1{}^2 = var(n_u(t))$, $\phi(x) = (1/2\pi)^{1/2}\,exp(-x^2/2)$, and $\Phi(x)$ is the probability distribution function of a zero mean and unit variance Gaussian random variable.

Another approximation commonly used is effectively to put the samples α_i through a baseband infinite clipper. This approximation reduced each term in the sum of Eq.(2.3) to a binary statistic and hence the final decision is made by majority logic decision rule. As indicated in several references ([1],[4],[5]), the final probability of error is a monotonic function of the probability of error of a single term of the sum of clipped samples and hence only the latter probability need be evaluated. Of the two techniques, the former (without the use of the clipper) is more realistic but the latter (using the clipper) is much simpler and yields qualitatively equivalent results.

III. ANALYSIS OF SYSTEM PERFORMANCE

Now, consider the evaluation of the probability of error of the two link system. Since the binary data are equiprobable, we can assume that a plus A is transmitted. From Eq.(2.3), then,

$$\begin{aligned} P_e &= P\{g(T) \le 0\} = P\{g_1(T) + g_2(T) \le 0\} \\ &= E\{\Phi(-g_1(T)/\sigma_2)\} \;, \end{aligned} \qquad (3.1)$$

where $\sigma_2{}^2$ is the variance of $g_2(T)$. Since $g_1(T)$ is used in the form of Eq.(2,4), it is consistent to use $g_2(T)$ in the form

$$g_2(T) \simeq \sum_{i=1}^{N} x_d(iT/N) \;.$$

Thus, the variance $\sigma_2{}^2 = N(2f_c n_{od})$, where $n_{od}/2$ is the two-sided power spectral density of the downlink noise. Then, by using Eqs.(2.4) and (2.5) in (3.1), we have

$$P_e = E\{\Phi(-\sum_{i=1}^{N} \alpha_i/\sigma_2)\} \;, \qquad (3.2)$$

where the α_i are all independent and identically distributed (i.i.d.) random variables with probability densities given by Eq.(2.6).

The P_e expression given in Eq.(3.2) can be evaluated by utilizing the moment space bounding technique [6]-[8]. Specifically, by using the isomorphism theorem, one establishes an equivalence between the convex hull of a curve generated by an arbitrary set of continuous kernel functions and the convex body generated by the generalized moments of these kernels. By proper selection of these kernels, the error probability can be identified as one of the moments, while other moments of the random variable of interest can be chosen such that they can be evaluated easily. Thus, upper and lower bounds to the error probability can be obtained from the envelopes of the convex body. A detailed statement and interpretation of the isomorphism theorem can be found in [6].

For our problem, we use

$$m_1 = P_e = E\{k_1(Y)\} \quad , \tag{3.3}$$

$$m_2 = E\{k_2(Y)\} \quad , \tag{3.4}$$

where

$$Y = \sum_{i=1}^{N} \alpha_i \tag{3.5}$$

$$k_1(y) = \Phi(-y/\sigma_2) \quad , \tag{3.6}$$

$$k_2(y) = \sum_{j=1}^{J} d_j \, exp(c_j y) \quad , \tag{3.7}$$

with $y \in [-N,N]$. By using Eqs.(3.5) and (3.7) in (3.4),

$$m_2 = \sum_{j=1}^{J} d_j \, E\{exp(c_j \sum_{i=1}^{N} \alpha_i)$$

$$= \sum_{j=1}^{J} d_j \prod_{i=1}^{N} E \, exp(c_j \alpha_i) \quad .$$

Since all α_i are i.i.d. random variables, we can let

$$M_j = E\{exp(c_j \alpha)\}$$

$$= \int_{-1}^{1} exp(c_j \alpha) f_\alpha(\alpha) d\alpha \quad .$$

Thus, m_2 can be expressed as

$$m_2 = \sum_{j=1}^{J} d_j (M_j)^N \quad . \tag{3.8}$$

In order to use Eqs.(3.3) and (3.4), we need to specify $\{c_j\}$ and $\{d_j\}$ explicitly. Since there is no known theory on selecting $\{c_j\}$ in the optimum minimax (Chebyshev) norm sense [8], we will simply choose

$$-N/\sigma_2{}^2 \le c_1 < c_2 \cdots < c_J \le N/\sigma_2{}^2 \ .$$

For fixed $\{c_j\}$, we can obtain $\{d_j\}$ by selecting $\{y_m, 1 \le m \le M\}$ to satisfy

$$-N \le y_1 < y_2 \cdots < y_M \le N \ ,$$

such that

$$k_1(y_m) = k_2(y_m) \ , \quad 1 \le m \le M \ . \tag{3.9}$$

By letting

$$\underline{k}_1 = [k_1(y_1), k_1(y_2), \cdots, k_1(y_M)]'$$

$$\underline{d} = [d_1, d_2, \cdots, d_J]'$$

$$\underline{e} = \begin{bmatrix} e^{c_1 y_1} & e^{c_2 y_1} & \cdots & e^{c_J y_1} \\ \vdots & & & \\ e^{c_1 y_M} & e^{c_2 y_M} & \cdots & e^{c_J y_M} \end{bmatrix}$$

Eq.(3.9) then becomes

$$\underline{k}_1 = \underline{e}\,\underline{d} \tag{3.10}$$

and $\underline{d}$ is obtained from

$$\underline{d} = \underline{e}^{-1}\,\underline{k}_1 \ . \tag{3.11}$$

Of course, since $\underline{e}$ is an $M \times J$ matrix, if $M \neq J$, then $\underline{e}^{-1}$ is a pseudo-inverse and the $\underline{d}$ given by Eq.(3.11) yields the optimum mean-square error criterion solution that satisfies Eq.(3.10). In practice, while J is restricted to be of moderate value in order not to cause a stability problem in the construction of $k_2(y)$ in Eq.(3.7), M is usually taken to be greater than J in order to obtain precise $\underline{d}$ values that satisfy Eq.(3.10). By using $\underline{d}$ given by (3.11) in (3.7), $k_2(y)$ then approximates $k_1(y)$ well over $[-N,N]$. Thus, by using $\underline{d}$ given by Eq.(3.11) in (3.8), m_2 given by (3.8) approximates the desired P_e closely.

IV. NUMERICAL RESULTS AND CONCLUSIONS

In Fig. 3, the error probabilities of a two-link system with equal uplink and downlink signal-to-noise ratios are given. Curves are presented for various values of the parameter ε/A, which is a measure of the degree of "hardness" of the limiter. These results are obtained with $N = 2f_cT = 1$. The $\varepsilon/A = -\infty$ dB curve represents the hard limiter result and is consistent with the result in Fig. 2 of [5]. As expected, the performance of this system improves with increasing amount of hardness of the limiter. In Fig. 4, results for $N = 2f_cT$ equal to one, two, and three with a hard limiter are given. As expected, for increasing N, the performance of the system decreases, since more random noise is allowed to reach the detector. In conclusion, we have presented an efficient computational technique for the evaluation of error probability of a two-link PSK communication system with bandpass non-linearity and Gaussian noise on both links. Extensions to other satellite system models are possible.

REFERENCES

[1] I. Jacobs, "The effects of video clipping on the performance of an active satellite PSK communication system," *IEEE Trans.Comm.Tech, COM-13* (June 1965) 195-201.

[2] L.D. Davisson and L.B. Milstein, "On the performance of digital communication systems with bandpass limiters. Part I: One-link systems," *IEEE Trans.Comm., COM-21* (Oct. 1972) 972-975.

[3] L.S. Davisson and L.B. Milstein, "On the performance of digital communication systems with bandpass limiters. Part II. Two link systems," *IEEE Trans.Comm., COM-21* (Oct. 1972) 975-980.

[4] R.G. Lyons, "The effect of a bandpass nonlinearity on signal detectability," *IEEE Trans.Comm., COM-21* (Jan. 1973) 51-60.

[5] P.C. Jain and N.M. Blachman, "Detection of a PSK signal transmitted through a band-limited channel," *IEEE Trans.Inform.Th., IT-19* (Sept.1973) 623-630.

[6] K. Yao and R.M. Tobin, "Moment space upper and lower error bounds for digital systems with intersymbol interference," *IEEE Trans.Inform.Th., IT-22* (1976) 65-75.

[7] R.M. Tobin and K. Yao, "Upper and lower error bounds for coherent phase-shift keyed (CPSK) systems with co-channel interference," *IEEE Trans. Comm., COM-25* (Feb. 1977) 281-287.

[8] K. Yao, "Error probability of asynchronous spread-spectrum multiple-access communication systems," *IEEE Trans.Comm., COM-25* (Aug. 1977) 803-809.

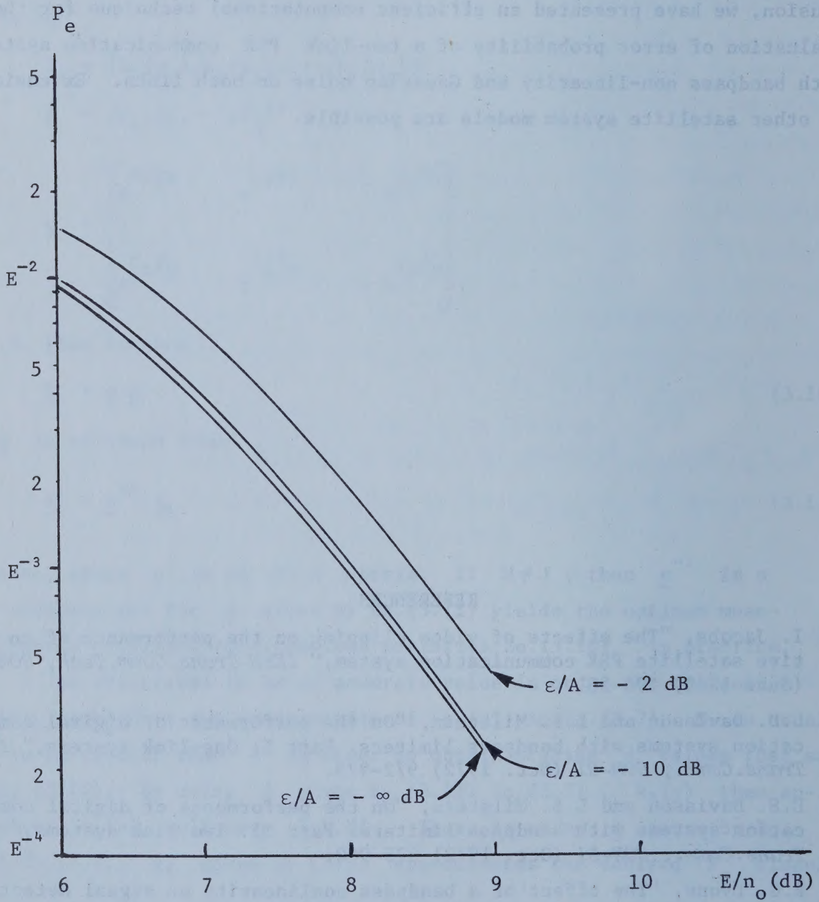


FIG. 3 Error Probability with $N=1$.

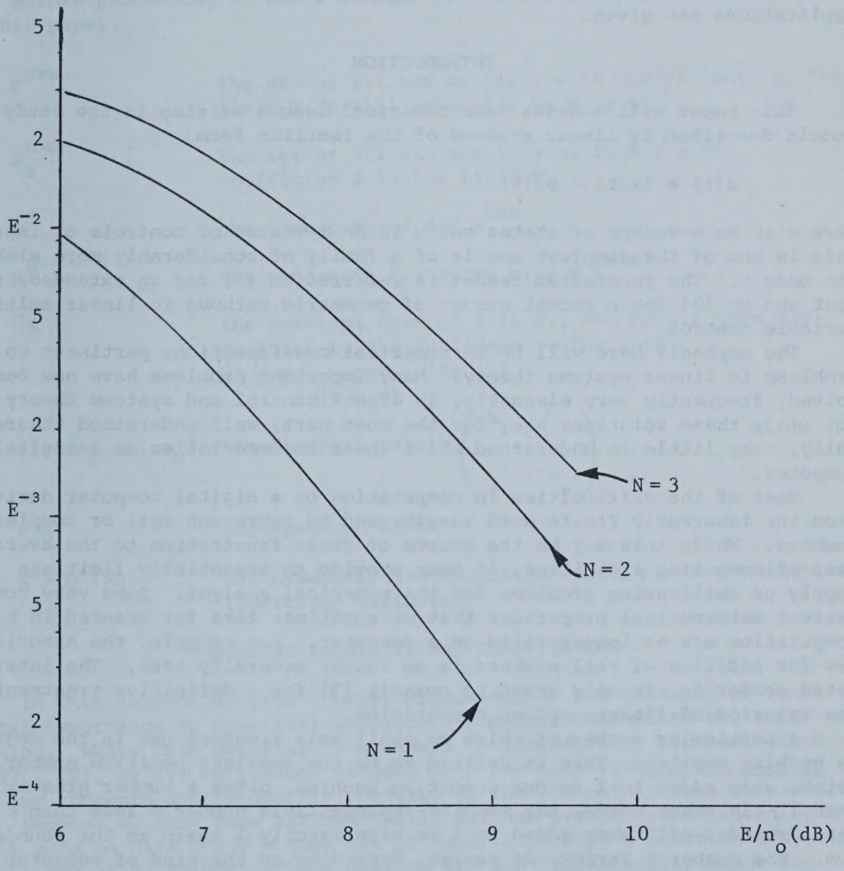


FIG. 4 Error Probability with Hard Limiter

COMPUTATIONAL ASPECTS OF THE SINGULAR VALUE DECOMPOSITION AND SOME APPLICATIONS*

ALAN J. LAUB
Laboratory for Information and Decision Systems, Room 35-331,
Massachusetts Institute of Technology, Cambridge, MA. 02139

ABSTRACT

This paper provides a brief tutorial introduction to certain numerical computations both in linear algebra and linear systems in the context of a finite word length computing environment. After a discussion of numerical stability and conditioning the singular value decomposition is presented along with some related comments on numerical rank. Finally, a variety of applications are given.

1. INTRODUCTION

This paper will address some numerical issues arising in the study of models described by linear systems of the familiar form

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{1}$$

Here x is an n-vector of states and u is an m-vector of controls or inputs. This is one of the simplest models of a family of considerably more elaborate models. The interested reader is referred to [1] for an extended treatment and to [2] for a recent survey of geometric methods in linear multivariable control.

The emphasis here will be on numerical considerations pertinent to problems in linear systems theory. Many important problems have now been solved, frequently very elegantly, in linear control and systems theory. But while these solutions are, for the most part, well-understood theoretically, very little is understood about their implementation on a digital computer.

Most of the difficulties in computation on a digital computer derive from the inherently finite word length used to represent real or complex numbers. While this may be the source of great frustration to the average user of computing facilities, it does provide an essentially limitless supply of challenging problems for the numerical analyst. Some very convenient methematical properties that we sometimes take for granted in hand computation are no longer valid on a computer. For example, the associative law for addition of real numbers is no longer generally true. The interested reader is strongly urged to consult [3] for a definitive treatment of the vagaries of floating point computation.

A particular number of which we shall make frequent use in the sequel is machine epsilon. This is defined to be the smallest positive number ε which, when added to 1 on our computing machine, gives a number greater than 1. In other words, any machine-representable number δ less than ε gets "rounded-off" when added to 1 to give exactly 1 again as the rounded sum. The number ε varies, of course, depending on the kind of computer being used and the precision with which the computations are being done (single precision, double precision, etc.). But the fact that there exists such a positive number ε is entirely a consequence of finite word length.

Explicit considerations of roundoff error, stability, conditioning, overflows and underflows, etc. have been heretofore largely ignored by the

---

* Research supported by Dept. of Energy, ERDA-E(49-18)-2087 and NASA/Ames Research Center, NGL-22-009-124.

control/systems community and research by that community into stable, efficient, and reliable algorithms and their embodiment in robust mathematical software is only now in its infancy. The time has arrived when "amateurs" can no longer, for many classes of problems, duplicate the efforts of numerical and software experts (for example [4], [5], [6]). Many problems in control and systems theory are of such a form as to directly benefit from those efforts now,while, conversely, many problems in control/systems suggest exciting new avenues of numerical and software research. While it is a slow process, we are now just beginning to see some of the material (well-known to numerical analysts) presented in this paper filter down to the undergraduate curriculum in mathematics and engineering. This process is certain to have a significant impact on the future directions and development of control and systems theory and applications.

Before proceeding we shall standardize here some notation to be used in this paper.

$\mathbb{F}^{n\times m}$	the set of all nxm matrices with coefficients in the field $\mathbb{F}$ ($\mathbb{F}$ will generally be $\mathbb{R}$ or $\mathbb{C}$)
$\mathbb{F}_r^{n\times m}$	the set of all nxm matrices of rank r with coefficients in the field $\mathbb{F}$
A^T	the transpose of $A \in \mathbb{R}^{n\times m}$
A^H	the conjugate transpose of $A \in \mathbb{C}^{n\times m}$
$\|A\|$	the spectral norm of A (i.e., the matrix norm subordinate to the Euclidean vector norm: $\|A\| = \max\limits_{\|x\|_2=1} \|Ax\|_2$)
$\mathrm{diag}(a_1,\ldots,a_n)$	the diagonal matrix $\begin{pmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{pmatrix}$
$\sigma(A)$	the spectrum of A
$A \geq 0$ [>0]	the symmetric (or Hermitian) matrix A is nonnegative [positive] definite

2. NUMERICAL STABILITY AND CONDITIONING

In this section we give a brief discussion of two concepts of fundamental importance in numerical analysis: numerical stability and conditioning. While this material is standard in introductory textbooks such as [7], [8], or [9] we present it here both for completeness and because the two concepts are frequently confused in the control/systems literature.

Suppose we have some mathematically defined problem represented by f which acts on data $d \in \mathscr{D}$ = some set of data,to produce a solution $f(d) \in \mathscr{S}$ = some set of solutions. These notions are kept deliberately vague for expository purposes. Given $d \in \mathscr{D}$ we desire to compute f(d). Frequently, only an approximation d* to d is known and the best we could hope for is to calculate f(d*). If f(d*) is "near" f(d) the problem is said to be well-conditioned. If f(d*) may potentially differ greatly from f(d) even when d* is near d, the problem is said to be ill-conditioned. Again the concept "near" cannot be made precise without further information about a particular problem.

Let f* denote the algorithm used to implement f. Given d, f*(d) represents the approximate computed solution. The algorithm f* is said to be numerically stable if for all $d \in \mathscr{D}$, there exists $d^* \in \mathscr{D}$ near d such that f(d*) (= the exact solution of a nearby problem) is near f*(d).

Of course, one can't expect a stable algorithm to solve an ill-conditioned problem any more accurately than the data warrant but an unstable algorithm can produce poor solutions even to well-conditioned problems. There are thus two separate factors to consider in determining the accuracy of a computed solution f*(d). First, if the algorithm is stable f*(d) is near f(d*) and second, if the problem is well-conditioned f(d*) is near f(d). Thus f*(d) is near f(d). Further details and examples may be found in [8].

Roundoff errors can cause unstable algorithms to give disastrous results. However, it would be virtually impossible to account for every roundoff error made at every arithmetic operation of a complex calculation. This would constitute a forward error analysis. To account for these errors, however, J. H. Wilkinson developed, to a fine degree, the notion of backward error analysis; see [3] and [10]. Namely, for many problems, it is possible to show that what is actually computed is the exact solution of a nearby problem. One then attempts to show that the nearby problem is near enough, which, if the problem is well-conditioned, can be translated into a quantitative statement regarding the accuracy of the solution.

For example, in the QR algorithm for finding the eigenvalues of a matrix A (implemented in [4], for example) it can be proved that the computed eigenvalues are the exact eigenvalues of the matrix A + E where $\|E\| \leq c \cdot \|A\| \cdot \varepsilon$ (c = a modest constant involving the order of the matrix, $\|\cdot\|$ = an appropriate matrix norm, ε = machine epsilon). This is the statement of stability for the QR algorithm.

3. SINGULAR VALUE DECOMPOSITION AND NUMERICAL RANK

One of the basic and most important tools of modern numerical analysis, particularly numerical linear algebra, is the singular value decomposition (SVD). The SVD was established for real square matrices in the 1870's by Beltrami and Jordan (see, e.g., MacDuffee [11], p.78), for complex square matrices by Autonne [12], and for general rectangular matrices by Eckart and Young [13] (the Autonne-Eckart-Young Theorem). For a survey of the theory for compact operators see Gohberg and Kreĭn [14]. We shall define the SVD here and make a few comments about its properties and computation. In section 4 we shall see how the SVD can be used to reliably compute a number of the basic geometric objects of linear algebra. This is but one of many fields of application and it is likely that within five or ten years SVD will be one of the most important and fundamental working tools for the control/systems community, particularly in the area of linear systems.

We now state the SVD theorem. Square brackets will denote the complex case.

THEOREM: Let $A \in \mathbb{R}_r^{n\times m}$ $[\mathbb{C}_r^{n\times m}]$. Then there exist orthogonal [unitary] matrices $U \in \mathbb{R}^{n\times n}$ $[\mathbb{C}^{n\times n}]$ and $V \in \mathbb{R}^{m\times m}$ $[\mathbb{C}^{m\times m}]$ such that

$$A = U\Sigma V^T \quad [U\Sigma V^H]$$

where $\Sigma = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}$ and $S = \mathrm{diag}(\sigma_1,\ldots,\sigma_r)$ with

$\sigma_1 \geq \ldots \geq \sigma_r > 0$.

PROOF: Since $A^TA \geq 0$ we have $\sigma(A^TA) \subseteq [0, +\infty)$.

Denoting $\sigma(A^TA)$ by $\{\sigma_i^2, i=1,\ldots,m\}$ we can arrange that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 = \sigma_{r+1} = \ldots = \sigma_m$. Let $v_1,\ldots,v_m$ be a corresponding set of orthonormal eigenvectors and let

$$V_1 = (v_1,\ldots,v_r)$$

$$V_2 = (v_{r+1},\ldots,v_m).$$

Then if $S = \mathrm{diag}(\sigma_1,\ldots,\sigma_r)$ we have $A^TAV_1 = V_1S^2$ whence

$$S^{-1}V_1^TA^TAV_1S^{-1} = I. \tag{2}$$

Also $A^TAV_2 = V_2 \cdot 0$ so that $V_2^TA^TAV_2 = 0$ and thus $AV_2 = 0$. Let $U_1 = AV_1S^{-1}$. Then from (2) we have $U_1^TU_1 = I$. Choose any U_2 such that $U = (U_1,U_2)$ is orthogonal. Then

$$U^TAV = \begin{pmatrix} U_1^TAV_1 & U_1^TAV_2 \\ U_2^TAV_1 & U_2^TAV_2 \end{pmatrix} = \begin{pmatrix} S & 0 \\ U_2^TU_1S & 0 \end{pmatrix} = \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} = \Sigma$$

and so $A = U\Sigma V^T$ as desired. ■

The numbers $\sigma_1,\ldots,\sigma_r$ together with $\sigma_{r+1} = 0,\ldots,\sigma_m = 0$ are called the singular values of A and they are the positive square roots of the eigenvalues of $A^TA[A^HA]$. The columns of U are called the left singular vectors of A (the orthonormal eigenvectors of AA^T $[AA^H]$) while the columns of V are called the right singular vectors of A (the orthonormal eigenvectors of $A^TA[A^HA]$). The matrix $A^T[A^H]$ has n singular values, the positive square roots of the eigenvalues of $AA^T[AA^H]$. The r (=rank (A)) nonzero singular values of A and $A^T[A^H]$ are, of course, the same. The choice of $A^TA[A^HA]$ rather than $AA^T[AA^H]$ in the definition of singular values is arbitrary.

Only the potentially nonzero singular values will be of any real interest.

It is not generally a good idea to compute the singular values of A by first finding the eigenvalues of A^TA, tempting as that is. Consider the following example ([15]) with μ a real number with $|\mu| < \sqrt{\varepsilon}$ (so that $fl(1 + \mu^2) = 1$ where $fl(\cdot)$ denotes floating point computation). Let $A = \begin{pmatrix} 1 & 1 \\ \mu & 0 \\ 0 & \mu \end{pmatrix}$. Then $fl(A^TA) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ so we compute $\hat{\sigma}_1 = \sqrt{2}$, $\hat{\sigma}_2 = 0$ leading to the (erroneous) conclusion that the rank of A is 1. Of course, if we could compute in infinite precision we would find $A^TA = \begin{pmatrix} 1+\mu^2 & 1 \\ 1 & 1+\mu^2 \end{pmatrix}$ with $\sigma_1 = \sqrt{2+\mu^2}$, $\sigma_2 = |\mu|$ and thus rank (A) = 2. The point is that by working with A^TA we have unnecessarily introduced μ^2 into the computations.

Fortunately, Golub and Reinsch [15] have developed an extremely efficient and stable algorithm (based largely on [16]) for computing the SVD which does not suffer from the above defect. The computed U and V are orthogonal to approximately the working precision and the computed singular values can be shown to be the exact σ_i's for A + E where $\frac{\|E\|}{\|A\|}$ is a modest multiple of ε. A fairly sophisticated implementation of this algorithm can be found in [5] (SVD and MINFIT). A word of warning! There are other SVD subroutines around and many have severe problems even if coded directly from [15]. One is probably best off using [5], the version implemented in the forthcoming LINPACK from Argonne National Laboratories [17], or the version in [9]. Using the SVD is more than just calling a subroutine, however, and the numerical analysis literature should be consulted before doing any serious numerical computations.

It is clear from the definition that the number of nonzero singular values of A determines its rank. While the question is not nearly as clear-cut in the context of computation on a digital computer, it is now generally acknowledged that the singular value dccomposition is the only generally reliable method of determining rank numerically (see [18] for a more elaborate discussion).

Only rather recently has the problem of numerical determination of rank been well-understood. One of the best treatments of the subject, including a careful definition of numerical rank, is a paper by Golub, Klema, and Stewart [19]. The essential idea is as follows. We are going to look at the "smallest nonzero singular value" of a matrix A. Since that computed value is exact for a matrix near A it makes sense to consider the rank of all matrices in some δ-ball (w.r.t. the spectral norm $\|\cdot\|$, say) around A. The choice of δ may also be based on measurement errors incurred in estimating the coefficients of A or the coefficients may be uncertain because of roundoff errors incurred in a previous computation to get them. See [19] for further details).

In any case it can easily be shown that all matrices B lying strictly inside the σ_r-ball around A have rank $\geq$ r. The matrix $B = U\hat{\Sigma}V^T$ where $\hat{\Sigma} = \begin{pmatrix} \hat{S} & 0 \\ 0 & 0 \end{pmatrix}$ with $\hat{S} = \operatorname{diag}(\sigma_1, \ldots, \sigma_{r-1})$ is a matrix with rank r-1 and

$\|B-A\| = \sigma_r$. Thus if we choose as some "zero threshold" a number $\delta < \sigma_r$, we will consider A to have numerical rank r. There can sometimes be real difficulties in determining a "gap" between the computed last nonzero singular value and what should effectively be considered "zero". Further details are found in [19].

The key quantity in rank determination is obviously σ_r. Moreover, this number gives a dependable measure of how far (in the $\|\cdot\|$ sense) a matrix is from matrices of lesser rank. But σ_r alone is clearly sensitive to scale so that a better measure is $\frac{\sigma_r}{\|A\|}$. But $\|A\| = \sigma_1$ so the important quantity is $\frac{\sigma_r}{\sigma_1}$ which turns out to be the reciprocal of the number $\kappa(A) = \|A\|\cdot\|A^+\|$, the so-called "condition number of A w.r.t. pseudo-inversion". In the case when A is invertible, $\kappa(A) = \|A\|\cdot\|A^{-1}\|$ is the usual spectral condition number w.r.t. inversion. The use of $\kappa(A)$ as a condition number in the general case is actually somewhat more complicated. For a discussion of this and related matters the reader is urged to consult a nice survey paper by Stewart [20].

In solving the linear system Ax=b, the condition number $\kappa(A) = \|A\|\cdot\|A^{-1}\|$ gives a measure of how much errors in A and/or b may be magnified in the computed solution. Moreover, if $A \in \mathbb{R}^{n\times n}$, $\sigma_n(A)$ gives a measure of the "nearness" of A to singularity. A standard introduction to the solution of linear systems is the classical book of Forsythe and Moler [21] to which we further refer the interested reader. Note, however, that in this special case of invertible A there is nothing special about the spectral norm and $\sigma_n(A)$. In fact $\frac{1}{\|A^{-1}\|}$ measures the nearness of A to singularity in any matrix norm $\|\cdot\|$ and for certain norms it is easy to construct explicitly a matrix E with $\|E\| = \frac{1}{\|A^{-1}\|}$ and A+E singular In the nonsquare case things are not quite so straightforward except in the spectral norm case.

That other methods of rank determination are potentially unreliable is demonstrated by the following special case of a family of matrices considered by Ostrowski [22]. The near-singularity properties of this special case (actually its negative) were emphasized by Kahan [23]. Consider the matrix $A \in \mathbb{R}^{n\times n}$ whose diagonal elements are all -1, whose upper triangle elements are all +1, and whose lower triangle elements are all 0. This matrix is clearly of rank n, i.e., is invertible. It has a good "solid" upper triangular shape. All of its eigenvalues (all = -1) are well away from zero. Its determinant is $(-1)^n$-- definitely not close to zero. But this matrix is, in fact, very near singular and gets more nearly so as n increases. Notice, for example, that

$$\begin{pmatrix} -1 & +1 & \ldots & +1 \\ & \ddots & \ddots & \vdots \\ & & \ddots & +1 \\ \bigcirc & & & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1/2 \\ \vdots \\ 1/2^{n-1} \end{pmatrix} = \begin{pmatrix} -1/2^{n-1} \\ -1/2^{n-1} \\ \vdots \\ -1/2^{n-1} \end{pmatrix} \to \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Moreover, adding $\frac{1}{2^{n-1}}$ to every element in the first column of A gives an exactly singular matrix. Arriving at such a matrix by, say Gaussian elimination, would give no hint as to the near-singularity. However, it is easy to check that $\sigma_n(A)$ (or $\frac{1}{\|A^{-1}\|}$ in any norm) behaves as $\frac{1}{2^n}$. A corollary for control theory: eigenvalues don't necessarily give a reliable measure of "stability margin".

Rank determination, in the presence of roundoff error, is a highly nontrivial problem. And, of course, all the same difficulties arise in any problem equivalent to or involving rank determination such as determining the independence of vectors, finding a basis for Ker A, etc. We turn now to some of these problems which naturally arise in linear multivariable control.

4. SOME APPLICATIONS OF SVD

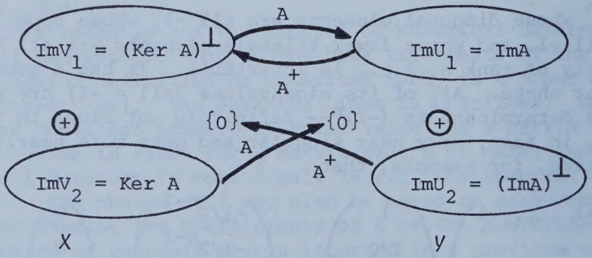
In this section we outline the application of singular value techniques to the computation of various subspaces and maps that arise in linear algebra and linear systems theory. No proofs will be given nor will the list of topics considered be exhaustive. Rather we shall attempt to impart only the flavor of singular value analysis to the subject, leaving technical matters and the details of software implementation for consideration elsewhere. Notation used will be fairly standard in linear algebra and is consistent with Wonham [1].

4.1 Four Fundamental Subspaces

Consider a linear map $A: \mathcal{X} \rightarrow \mathcal{Y}$ between two finite-dimensional real vector spaces $\mathcal{X}$ and $\mathcal{Y}$ with $d(\mathcal{X}) = m$, $d(\mathcal{Y}) = n$. Identifying A with an nxm real matrix A suppose A has an SVD given by

$$A = U\Sigma V^T = (U_1, U_2) \begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$$

where $S = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \ldots \geq \sigma_r > 0$ and U and V are partitioned compatibly (U_1 is n x (n-r), etc.) Then the U_i and V_i provide orthonormal bases for the four fundamental subspaces in the following self-explanatory diagram:


A
$\mathrm{Im}V_1 = (\mathrm{Ker}\, A)^{\perp}$
$\mathrm{Im}U_1 = \mathrm{Im}A$
A^+
$\oplus$
{0}
{0}
$\oplus$
A
A^+
$\mathrm{Im}V_2 = \mathrm{Ker}\, A$
$\mathrm{Im}U_2 = (\mathrm{Im}A)^{\perp}$
$\mathcal{X}$
$\mathcal{Y}$

As discussed in Section 2 we have the computational problem of intelligently deciding what is σ_r and hence the rank of A. But that decision directly affects each of the above subspaces and hence their calculation. Since SVD is the only generally reliable way of calculating rank it follows that it is the only generally reliable way of calculating bases for these subspaces.

4.2 Projections

The four fundamental orthogonal projections are given by:

$$U_1U_1^T = P_{\mathrm{Im}A} = AA^+$$

$$U_2U_2^T = P_{\mathrm{Ker}A^T} = I-AA^+$$

$$V_1V_1^T = P_{\mathrm{Im}A^T} = A^+A$$

$$V_2V_2^T = P_{\mathrm{Ker}A} = I-A^+A$$

For the case of oblique projections, suppose $\mathrm{Im}R = \mathcal{R}$ and $\mathrm{Im}S = \mathcal{S}$ where $\mathcal{R}$ and $\mathcal{S}$ are two subspaces of $\mathcal{X}$ such that $\mathcal{R} \oplus \mathcal{S} = \mathcal{X}$. Notice that R and S need not have linearly independent columns. Then with the obvious choice of notation we have for the projection on $\mathcal{R}$ along $\mathcal{S}$

$$P_{\mathcal{R},\mathcal{S}} = (U_{1R},0)(U_{1R},U_{1S})^{-1}$$

with similar expressions for $P_{\mathcal{S},\mathcal{R}}$, $P_{\mathcal{R},\mathcal{S}^\perp}$, etc. Notice that in the special case of $\mathcal{S} = \mathcal{R}^\perp$ we do indeed have

$$P_{\mathcal{R},\mathcal{R}^\perp} = (U_{1R},0)(U_{1R},U_{2R})^{-1}$$

$$= (U_{1R},0)\begin{pmatrix} U_{1R}^T \\ U_{2R}^T \end{pmatrix}$$

$$= U_{1R}U_{1R}^T = RR^+ = P_{\mathrm{Im}R} = P_{\mathcal{R}}.$$

4.3 The Factoring of a Linear Map

Suppose $A\colon \mathcal{X} \to \mathcal{Y}$, $\mathcal{R} \subseteq \mathrm{Ker}\, A \subseteq \mathcal{X}$ (here $\subseteq$ denotes "is a subspace of"). Let $P\colon \mathcal{X} \to \dfrac{\mathcal{X}}{\mathcal{R}}$ be the canonical projection. Then there exists a unique linear map $Q\colon \dfrac{\mathcal{X}}{\mathcal{R}} \to \mathcal{Y}$ such that the following diagram commutes

$$\begin{tikzcd} \mathcal{X} \arrow[r, "P"] \arrow[dr, "A"'] & \dfrac{\mathcal{X}}{\mathcal{R}} \arrow[d, dashed, "Q"] \\ & \mathcal{Y} \end{tikzcd}$$

To compute P and Q, suppose $\mathrm{Im}R = \mathcal{R}$ and let R have SVD

$$R = (U_1,U_2)\begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}.$$

Then P is given by $P = U_2^T$ while the induced Q is given by $Q = AU_2$. Note that $\mathrm{Ker}\, P = \mathcal{R} \subseteq \mathrm{Ker}\, A$ is equivalent to $ARR^+ = 0$ whence $A = A(I-RR^+) = AU_2U_2^T = QP$. In the case when $\mathscr{R} = \mathrm{Ker}\, A$ we have $AU_1 = 0$ and $A = QP = (AU_2) \cdot U_2^T$ factors A into the product of injective and surjective maps.

4.4 The Induced Map in the Factor Space

Suppose A: $\mathcal{X} \to \mathcal{X}$, and $\mathcal{R} \subseteq \mathcal{X}$ is A-invariant, i.e., $A\mathcal{R} \subseteq \mathcal{R}$. Then A induces a unique endomorphism $\bar{A}$ of the factor space $\frac{\mathcal{X}}{\mathcal{R}}$ which makes the following diagram commute

$$\begin{tikzcd} \mathcal{X} \arrow[r, "A"] \arrow[d, "P"'] & \mathcal{X} \arrow[d, "P"] \\ \frac{\mathcal{X}}{\mathcal{R}} \arrow[r, dashed, "\bar{A}"] & \frac{\mathcal{X}}{\mathcal{R}} \end{tikzcd}$$

P = the canonical projection

Again suppose R is any matrix with ImR = $\mathcal{R}$ and let R have SVD $R = (U_1, U_2)\begin{pmatrix} S & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix}$. Then P is given by U_2^T while $\bar{A}$ is given by $\bar{A} = U_2^T A U_2$. Note that $A\mathcal{R} \subseteq \mathcal{R}$ is equivalent to

$$RR^+AR = AR$$

which implies $U_2^TAU_1 = 0$ and $U_2^TA(I-U_2U_2^T) = 0$. Thus $\bar{A}P = U_2^TAU_2U_2^T = U_2^TA = PA$. Also note that under the orthogonal change of basis U, A becomes

$$U^TAU = \begin{pmatrix} U_1^TAU_1 & U_1^TAU_2 \\ 0 & U_2^TAU_2 \end{pmatrix} = \begin{pmatrix} A|\mathcal{R} & U_1^TAU_2 \\ 0 & \bar{A} \end{pmatrix} .$$

4.5 Subspaces

Suppose $\mathcal{R} \subseteq \mathcal{X}$ and A: $\mathcal{X} \to \mathcal{X}$. Let R be any matrix with ImR = $\mathcal{R}$ and let the SVD of R be as above. Then $d(\mathcal{R})$ = the number of nonzero singular values of R and the columns of U_1 give an orthonormal basis for $\mathcal{R}$ while the columns of U_2 give an orthonormal basis for $\mathcal{R}^\perp \approx \left(\frac{\mathcal{X}}{\mathcal{R}}\right)'$. A basis for $A\mathcal{R}$ can be obtained from the U_1 of the SVD of AR. To get a basis for $A^{-1}\mathcal{R}$ we need the SVD of A. Then

$$\begin{aligned} A^{-1}\mathcal{R} &= \text{Im } A^+R + \text{Ker } A \\ &= \text{Im}[V\Sigma^+U^TR,\ V_2] \\ &= \text{Im}[V_1S^{-1}U_1^TR,\ V_2] \end{aligned}$$

4.6 The Calculus of Subspaces

Given two subspaces $\mathcal{R}$ and $\mathcal{T}$ of $\mathcal{X}$ it must frequently be determined if $\mathcal{R} \subseteq \mathcal{T}$. Suppose ImR = $\mathcal{R}$ and ImT = $\mathcal{T}$. Then $\mathcal{R} \subseteq \mathcal{T}$ if and only if $TT^+R = R$ so using U_1 from the SVD of T one can check if $U_1U_1^TR = R$. Alternatively (but more expensively) one can check if the number of nonzero singular values of T equals the number of nonzero singular values of the matrix

[R,T]. To check if $R = T$ both $R \subseteq T$ and $T \subseteq R$ can be verified.

Upon computing the SVD of [R,T], the columns of U_1 are an orthonormal basis for $R + T$ while the columns of U_2 are an orthonormal basis for $R^{\perp} \cap T^{\perp}$. The columns of V_1 and V_2 do not appear to span anything particularly interesting. The extension of this procedure to arbitrary finite sums of subspaces is obvious.

The "dual calculation" of $R \cap T$ is not quite so easy. One can use $R \cap T = (R^{\perp} + T^{\perp})^{\perp}$ and do two SVD's to get bases for $R^{\perp}$, $T^{\perp}$ then proceed as above with one further SVD. This procedure extends obviously to arbitrary finite intersections of subspaces but is clearly expensive. An alternate procedure (but only for the case of two subspaces) is to do an SVD of [R,T]. Then from the partitioning

$$0 = [R,T]V_2 = [R,T]\begin{pmatrix} V_{2R} \\ V_{2T} \end{pmatrix}$$

choose RV_{2R} or TV_{2T} (according as $d(R)$ or $d(T)$ is smaller) as the appropriare basis.

4.7 Other Applications

There are numerous other important applications of the singular value decomposition when quantative and qualitative information is desired about a linear map. One of the major areas is in the solution of linear least squares problems. The book by Lawson and Hanson [24] is a good introduction to the subject but the subsequent numerical analysis literature should also be consulted for important recent developments; for example, [20] contains a nice survey of the perturbation theory.

The SVD also finds application in: (i) solving systems of linear equations with equality constraints; (ii) solving homogeneous systems of linear equations; (iii) computing the Moore-Penrose pseudoinverse and other generalized inverses; (iv) determining dependencies or near- dependencies among the columns or rows of a matrix (note particularly [19]).

The SVD also finds use as a tool for theoretical analysis of numerical algorithms and we would cite [18] and [25] as examples.

It suffices to say, in conclusion, that singular value analysis forms a cornerstone of modern numerical linear algebra and we are now just beginning, imperceptibly, to see its legitimate use in linear control and systems theory.

5. REFERENCES

1. Wonham, W. M., Linear Multivariable Control: A Geometric Approach, Springer-Verlag, New York, 1974.
2. Wonham, W. M., Geometric Methods in the Structural Synthesis of Linear Multivariable Controls, Proc. 1977 Joint Automatic Control Conf., San Francisco, CA., June 1977, pp. 594-600.
3. Wilkinson, J. H., Rounding Errors in Algebraic Processes, Prentice-Hall, Englewood Cliffs, N. J., 1963.
4. Smith, B. T. , et al, Matrix Eigensystem Routines--EISPACK Guide, Second Edition, Lect. Notes in Comp. Sci., Vol. 6, Springer-Verlag, N. Y., 1976.
5. Garbow, B. S., et al., Matrix Eigensystem Routines--EISPACK Guide Extension, Lect. Notes in Comp. Sci., Vol. 51, Springer-Verlag, N. Y., 1977.

6. Cowell, W., *Portability of Numerical Software*, Oak Brook, Ill., Lecture Notes in Computer Science, Vol. 57, Springer-Verlag, N. Y., 1977.
7. Dahlquist, G.,and Å. Björck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J., 1974.
8. Stewart, G. W., *Introduction to Matrix Computations*, Academic Press, New York, 1973.
9. Forsythe, G. E., M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, N. J., 1977.
10. Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.
11. MacDuffee, C.C., *The Theory of Matrices*, Springer, Berlin, 1933.
12. Autonne, L., Bull. Soc. Math. France, 30 (1902), pp. 121-133.
13. Eckart, C., and G. Young, A Principal Axis Transformation for Non-Hermitian Matrices, Bull.Amer. Math. Soc., 45 (1939), pp. 118-121.
14. Gohberg, I. C. and Kreĭn, M. G., Introduction to the Theory of Linear Non-Self-Adjoint Operators, Amer. Math. Soc., Providence, R. I., 1969.
15. Golub, G. H., and C. Reinsch, Singular Value Decomposition and Least Squares Solutions, Numer. Math., 14 (1970), pp. 403-420.
16. Golub, G. H.,and W. Kahan, Calculating the Singular Values and Pseudo-Inverse of a Matrix, SIAM J. Numer. Anal., 2 (1965), pp. 205-224.
17. Dongarra, J. J., et al., LINPACK Working Note #9, Preliminary LINPACK User's Guide, Argonne National Lab., Appl. Math. Div., TM-313, August 1977.
18. Golub, G. H.,and J. H. Wilkinson, Ill-Conditioned Eigensystems and the Computation of the Jordan Canonical Form, SIAM Rev., 18 (1976), pp. 578-619.
19. Golub, G. H., V. C. Klema, and G. W. Stewart, Rank Degeneracy and Least Squares Problems, Tech. Rept. STAN-CS-76-559, Computer Science Dept., Stanford University, Aug. 1976.
20. Stewart, G. W., On the Perturbation of Pseudo-Inverses, Projections, and Linear Least Squares Problems, SIAM Rev., 19 (1977), pp. 634-662.
21. Forsythe, G. E.,and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, N. J., 1967.
22. Ostrowski, A. M., On the Spectrum of a One-Parametric Family of Matrices, J. reine angew. Math., Bd. 193 (1954), pp. 143-160.
23. Kahan, W., Numerical Linear Algebra, Canad. Math. Bull., 9 (1966), pp. 756-801.
24. Lawson, C. L., and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N. J., 1974.
25. Cline, A. K., C. B. Moler, G. W. Stewart, and J. H. Wilkinson, An Estimate for the Condition Number of a Matrix, LINPACK Working Note #7, Argonne National Laboratory, Appl. Math. Div., TM-310, July 1977.

COMPUTATIONAL PROBLEMS WITH MODAL ANALYSIS

B.C. MOORE
Dept. of Electrical Engineering
University of Toronto
Toronto, Ontario M5S 1A4

ABSTRACT

In this paper recent results involving singular valve analysis of linear systems [1], [2] are used to expose practical problems with modal analysis. An intuitively appealing concept of a "nearly uncontrollable system" is developed and is shown to be inconsistent with the seemingly natural idea of a "nearly uncontrollable mode". Briefly stated, a system may be nearly uncontrollable with no corresponding small coefficients in the modal expansion. Illustrative examples are given.

I. INTRODUCTION

The purpose of this paper is to illustrate that there is a rather subtle trap associated with using the theory of modal analysis as a set of guidelines for applications work, where data and models are imprecise: Interpretations which seem logical and natural may be incorrect or misleading. The central theme of the paper is closely related to the vague notions of "a nearly uncontrollable system" and "a mode which is nearly uncontrollable". Recently developed tools for applied analysis of linear systems [1], [2], [3] will be briefly described and used with little elaboration.

Section 2 contains a brief (geometric) review of controllability and its role in minimal realization theory, together with a discussion of the modal version of these results. In section 3 the needed results from [1], [2] are given, and the idea of "near uncontrollability" is discussed. Section 4 contains examples and discussion to illustrate a problem which may be loosely stated as follows: The seemingly natural notion of a "nearly uncontrollable mode" is not consistent with that of a "nearly uncontrollable system" as developed in section 3.

The problem with modal analysis does not occur unless the eigenvalues of the system are "clustered", i.e. can be divided into subsets of eigenvalues grouped together in the complex plane. Although clustering is apparent in the examples given here, this is not typical. Only for very low order systems is it possible to determine whether a clustering problem exists by simply inspecting the eigenvalues of A. (see [4])

---

* Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada M5S 1A4.

This work was supported by NRC grant #A3925.

II. BACKGROUND

Consider the single input system

$$\dot{\underline{x}}(t) = A\underline{x}(t) + \underline{b}u(\underline{t}) \qquad (1)$$

where $\underline{x}(t) \in \mathbb{R}^n$, $\underline{u}(t) \in \mathbb{R}$, and $\underline{x}(0)=0$. We shall assume for simplicity, that the system is asymptotically stable.

First let us review the concept of controllability and its role in minimal realization theory. The state space can be decomposed into two parts; X_c, the controllability subspace, and $X_c^{\perp}$, its orthogonal complement. The controllability supspace is characterized most directly as,

$$X_c = \{\underline{x} \varepsilon \mathbb{R}^n : x = \int_0^{t_1} e^{A(t_1-\tau)} \underline{b}\ u(\tau)\ d\tau,\ t_1 < \infty, u(\cdot) \varepsilon C[0,t,]\}$$

where $C[0,t_1]$ represents the space of piecewise continuous functions on $[0,t_1]$. The space $X^{\perp}$ represents, geometrically, that part of the system which is "disconnected" from the input.

To display the "disconnection" from the input, we may let U_1, U_2 be matrices whose columns form orthonormal bases for X_c, $X_c^{\perp}$, respectively. Then with $\underline{z}_1(t) \triangleq U_1{}' \ \underline{x}(t)$ and $\underline{z}_2(t) \triangleq U_2{}' \ \underline{x}(t)$ we have, for $\underline{x}(0) = 0$,

$$\dot{z}_1(t) = \bar{A}_1 z_1(t) + \underline{\bar{b}}_1 \underline{u}(t)$$

$$z_2(t) = 0.$$

where $\bar{A}_1 = U_1{}' AU_1$ and $\underline{\bar{b}}_1 = U_1{}'\underline{b}$. The subsystem defined by $(\bar{A}_1, \underline{b}_1)$ is often called the controllable part of the system.

Now let us assume that A has distinct eigenvalues, and consider the modal version of controllability, which is based on the following result:
Let $\lambda_1, \lambda_2, \cdots \lambda_n$ represent the eigenvalues of A with corresponding unit eigenvectors $\underline{v}_1, \underline{v}_2, \underline{v}_n$, and define $V \triangleq (\underline{v}_1 \ \underline{v}_2 \ \cdots \ \underline{v}_n)$. With $\underline{x}(0) = 0$ and $\underline{b}$ expressed as a linear combination of the eigenvectors,

$\underline{b} = \sum_{i=1}^{n} \alpha_i \underline{v}_i$, the solution to (1) can be written,

$$\underline{x}(t) = \sum_{i=1}^{n} \alpha_i \underline{v}_i \int_0^t e^{\lambda_i (t - \tau)} u(\tau)\ d\tau \qquad (2)$$

It is clear that it is possible to excite the i^{th} term through the input if and only if $\alpha_i \neq 0$, in which case the i^{th} term is said to be a controllable mode. Moreover, it is not difficult to see that the set of eigenvectors corresponding to the controllable modes is a basis for X_c.

The relationships given in the previous paragraph imply that one can compute a basis for X_c by determining the subset of $\{\alpha_1, \alpha_2, \ldots \alpha_k\}$ consisting of those coefficients which are not equal to zero and retaining the corresponding eigenvectors. In principle, one may identify the subset of nonzero coefficients in one of three ways:

T1: Compute $\underline{\alpha} = V^{-1}\underline{b}$ and inspect its components.

T2: Compute $\hat{\alpha}_i = \underline{p}_i{}'\underline{b}$ where $\underline{p}_i$ satisfies $p_i^H p_i = 1$, $A'\underline{p}_i = \lambda_i \underline{p}_i$ (i.e. $\underline{p}_i$ is a unit left eigenvector corresponding to λ_i).
It can be shown that $\hat{\alpha}_i = 0 \Leftrightarrow \alpha_i = 0$

T3: Compute the singular values of $(A-\lambda_i I \quad \underline{b})$. It can be shown that $\alpha_i = 0 \iff$ there are fewer than n nonzero singular values.

Although minimal realization theory offers crucial insight, its application is hampered by the fact that controllability is a generic property; i.e. it is almost always true that $X_c = \mathbb{R}^n$. Hence it is almost always true that the coefficients $\alpha_1, \alpha_2, \ldots \alpha_n$ given in the modal expansion (2) are all nonzero. While it seems natural to use the relative magnitudes of the coefficients as an indication of the degree to which each mode is excited, this can lead to problems. Although it is true that modes with very small coefficients contribute little to the state response, the converse is not true. Individual modes with relatively large coefficients may act together in a way which is closely approximated by fewer modes.

III. NEAR UNCONTROLLABILITY

To provide a framework for the discussion of modal analysis and the relative importance of the dynamics of subsystems, we shall introduce an intuitively appealing idea of what is meant by a "nearly uncontrollable system". A more thorough discussion may be found in [1], [2], [3].

Suppose the state variables are in close correspondence with physical variables (pressure, temp., flow rates, etc.) and are scaled on a "per unit" basis so that if in (1) it happens that $\int_0^\infty x_i^2(t)\,dt$ is relatively small for some input $\underline{u}(t)$, we may conclude that the associated physical variable is only slightly excited by that input. Controllability theory says that there may exist an equivalent vector of state variables $\underline{z}(t) = \begin{pmatrix} \underline{z}_1(t) \\ \underline{z}_2(t) \end{pmatrix}$ where $\int_0^\infty \|\underline{z}_2(t)\|^2\,dt = 0$ (for $\underline{x}(0) = \underline{z}(0) = 0$). Realistically, in view of the fact that small perturbations destroy controllability structure, the most we can expect to find is a subset for which $\int_0^\infty \|\underline{z}_2(t)\|^2\,dt$ is relatively small. The meaning of "relatively small" will be made more precise in the next paragraph.

In [1], [2] it is shown that the set

$$S \triangleq \{\underline{x} \in \mathbb{R}^n : \underline{x} = \int_0^{t_1} e^{A(t_1-\tau)}\,\underline{b}u(\tau)\,d\tau,\ \int_0^t \|\underline{u}(t)\|^2\,dt = 1,$$

$$t_1 < \infty,\ \underline{u}(\cdot) \in C\,[0,t_1] \quad \}$$

is an ellipsoid with orthogonal axis aligned with the singular vectors of W, the symmetric square root of

$$W^2 = \int_0^\infty e^{At}\underline{b}\,\underline{b}'\,e^{A't}\,dt.$$

The axis lengths (Euclidian norm) are equal to the corresponding singular values of W. In other words, suppose the singular value decomposition of W gives $U\Sigma U'$, partitioned as

$$(U_1 U_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} U_1' \\ U_2' \end{pmatrix},$$

where $\Sigma_1 = \text{diag.}\{\sigma_1, \sigma_2, \cdots, \sigma_k\}$, $\Sigma_2 = \text{diag.}\{\sigma_{k+1} \cdots \sigma_n\}$

and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$.

Then

$$\int_0^{t_1} \|U_2'\,\underline{x}(t)\|^2\,dt \leq \sigma_{k+1}^2$$

for all $t_1 < \infty$ and input functions satisfying $\int_0^{t_1} |u(t)|^2\,dt = 1$.

Since U is orthogonal, $\underline{z}(t)$ retains the norm characteristics of $\underline{x}(t)$, and it seems reasonable to interpret the condition

$\sigma_{k+1} << \sigma_k$ to mean that the (linear combination of) physical variables corresponding to $\underline{z}_2(t)$ are only slightly excited by the input variable.

In geometric terms, we may decompose the state space into $X_s = Im\ (U_1)$ and $X_w = X_s^{\perp} = Im\ (U_2)$ (the subscripts "s" and "w" are intended to suggest "strong" and "weak"). The condition $\int_0^{t_1} ||\underline{u}(t)||^2\, dt = 1$ implies that the projection of $\underline{x}(t)$ on $X_s^{\perp}$ (call this $\underline{x}_s(t)$) satisfies

$$\int_0^t \| \underline{x}_s(t) \|^2\, dt \leq \sigma_{k+1}^2 .$$

If σ_{k+1} is very small, we may treat X_s, X_w, as "working" values of X_c, $X_c^{\perp}$.

The examples given in the next section show that if the matrix A has clustered eigenvalues, then "near uncontrollability" (as discussed in the preceeding paragraphs) may occur with no corresponding small coefficients in the modal expansion; i.e. no α_i's which are near zero.

IV EXAMPLES

Example 1:

The first example illustrates that eigenvalue clustering may cause the eigenvectors of A to be severly skewed, in which case they provide a poor set of basis vectors for X_c. Let

$$\dot{\underline{x}}(t) = \begin{pmatrix} -.1 & 1 & 0 \\ 0 & -.11 & 0 \\ 0 & 0 & -3 \end{pmatrix} \underline{x}(t) + \begin{pmatrix} 1 \\ .01 \\ 1 \end{pmatrix} u(t).$$

The eigenvectors corresponding to $\{-.1, -.11, -3\}$ are respectively*,

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -.9995 \\ -.0099995 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

and the response is

$$e^{At}\underline{b} = 2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} e^{-.1t} + \begin{pmatrix} .9995 \\ -.009995 \\ 0 \end{pmatrix} e^{-.11t} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} e^{-3t}.$$

---

* Computed numbers are rounded to five significant figures.

One can see by inspection that the first two modes act together (the two eigenvectors are nearly colinear and the modes decay at roughly the same speed) and might be well approximated by one mode with an appropriate time constant. In fact one might guess that

$$\begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} e^{-.105t} \qquad -\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} e^{-3t} \tag{3}$$

is a good (possibly not best) second order approximation of the impulse response matrix.

Applying the three tests for controllability from Section 2, one gets

T1: $\underline{\alpha} = V^{-1}\underline{b} = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}$

T2: $\hat{\underline{\alpha}} = \begin{pmatrix} .019999 \\ -.01 \\ -\ 1 \end{pmatrix}$

T3: $(A-\lambda_1 I \mid \underline{b})$ and $(A-\lambda_2 I \mid \underline{b})$ each have one small singular value.

The first test gives no small coefficients (as one would expect from the modal expansion), and tests T2, T3 give small coefficients corresponding to λ_1, λ_2. None of the tests indicate that the first two modes might be approximated by one mode.

Singular value analysis, as described in Section 2, gives $W = U \Sigma U'$ where

$$U = \begin{pmatrix} .998 & -.0604 & .00894 \\ .00904 & .00139 & -.9999 \\ .0604 & .998 & .00193 \end{pmatrix} ; \quad \Sigma = \begin{pmatrix} 2.35 & 0 & 0 \\ 0 & .384 & 0 \\ 0 & 0 & -.00179 \end{pmatrix}$$

The short axis of the ellipse (the one with length equal to .00179) is closely aligned with the vector (0 1 0)′, which is orthogonal to the controllable subspace of the second order approximation given by (3).

Example 2:

This example illustrates that problems may arise even if the eigenvectors are not highly skewed. Consider the system

$$\dot{\underline{x}}(t) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1.01 & 0 \\ 0 & 0 & 0.1 \end{pmatrix} \underline{x}(t) + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} u(t).$$

It is again clear that the first two modes act as one

$$e^{At}\underline{b} = \begin{pmatrix} e^{-t} \\ e^{-1.01t} \\ e^{-.1t} \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} e^{-t} \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} e^{-.1t}. \qquad (4)$$

Controllability tests T1, T2, give identical results, $\underline{\alpha}$, = $\hat{\underline{\alpha}}$ = (1 1 1)´, with no small coefficients. Test T3 gives a small singular value associated with each of the first two eigenvalues.

Singular value analysis gives $W = U \Sigma U'$ where

$$U = \begin{pmatrix} .199 & -.68 & -.706 \\ .197 & -.68 & .708 \\ .960 & .281 & .000778 \end{pmatrix}; \quad \Sigma = \begin{pmatrix} 2.32 & 0 & 0 \\ 0 & .788 & 0 \\ 0 & 0 & .00203 \end{pmatrix}$$

The short axis is closely aligned with (-1 1 0)´ which is orthogonal the controllable subspace of the approximation given by (4).

V CONCLUDING REMARKS

To summarize the problem with modal analysis: In the presence of clustering, which is not well defined or easily detected, individual inspection of the modes may fail to indicate groups of modes (of roughly the same speed) which act together.

The discussion of "near uncontrollability" given in Section 2, together with the two examples suggests a natural model reduction scheme: Let U_1 be a matrix whose columns form an orthonormal basis for X_s the working controllable subspace. Then

$$\underline{\dot{z}} = U_1{}' AU\underline{z} + U_1{}'\underline{b}\ u(t)$$

should be a reasonable model. This idea is pursued carefully in [2] and is discussed in a more narrative style in [3].

REFERENCES

1. B. C. Moore, "Singular Value Analysis of Linear Systems, PartI: Internal Variables", Systems Control Report No.7801,Dept. of Elect. Engr., University of Toronto, Toronto, Ontario,CANADA, M5S 1A4

2. ————, "Singular Value Analysis of Linear Systems, Part II: Controllability, Observability, and Model Reduction", Systems Control Report No. 7802, Dept. of Elect. Engr., University of Toronto, Toronto, Ontario, CANADA, M5S 1A4

3. ————, "Singular Value Analysis of Linear Systems", To be presented at the Seventeenth IEEE Conference on Decision and Control, San Diego, Calif., January, 1979.

4. G.H. Golub and J.H. Wilkinson, "Ill-Conditioned Eigensystems and the Computation of the Jordan Canonical Form", SIAM Review, 18 (1976),pp.578-619

TIGHT BOUNDS ON THE RESPONSE OF MULTIVARIABLE SYSTEMS WITH COMPONENT UNCERTAINTY[*]

MICHAEL G. SAFONOV
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, California 90007

ABSTRACT

For Multi-Input-Multi-Output (MIMO) feedback systems having internal components that are subject to parameter variations, nonlinearity, or unmodeled dynamics within given 'conic sector' bounds, tight bounds (which also turn out to take the form of 'conic sector' bounds) are derived for the input-output relation of the closed-loop system. In the case of linear-time invariant systems, the 'conic sector' bounds are interpretable in terms of the maximal singular values of certain frequency-response matrices.

INTRODUCTION

One of the main motivations for the use of feedback in control system design is to reduce the effects of plant ignorance on system response. In the case of single-input-single-output (SISO) feedback systems in which plant ignorance limited to a few (typically, 1 or 2) uncertain parameters known to lie within specified bounds, the Evans root-locus method [1] and Nichols chart frequency-response method [2, p. 197] are very useful (cf. [3, Ch. 6],[4]). In both cases, one graphically constructs 'regions of uncertainty' containing (for all possible parameter values) the representation of the system response in the root-locus plane or on the Nichol's chart. In the case of root locus approach, these 'regions' are the set of values assumed by each of the system poles as system parameters vary over specified ranges. Similarly, in the Nichol's chart approach these 'regions' are the set of values assumed by the systems open-loop log magnitude versus phase plot at each frequency as system parameters vary over specified ranges. Because the 'regions of uncertainty used in these approaches are **difficult to** construct for systems with more than one or two parameters and because these approaches are incompatible with MIMO systems or with systems having nonlinear or time-varying plant ignorance, there is a need for alternative approaches to the problem of bounding the effects of plant uncertainty on closed-loop system response. The results of this paper establish that 'conic sector conditions' provide such an alternative approach. For, linear time-invariant systems, these conditions can be expressed in terms of the 'singular values' of certain frequency response matrices.

NOTATION AND PROBLEM FORMULATION

We consider the class of systems which can be represented as a linear time-invariant (LTI) interconnection of N uncertain multi-input-multi-output components. It is assumed that an 'approximate' LTI model $\underset{\sim}{C}_i$ (i = 1,...,N) and associated transfer function matrix $\underline{C}_i(s)$ is available for each of the N uncertain components, but that the actual system components

---

[*] Research supported in part by NSF Grant ENG 78-05628, in part by Joint Services Electronics Program contract F44620-76C-0061 monitored by AFOSR, and in part by the Honeywell Systems and Research Center, Minneapolis, MN 55413 with support from the Office of Naval Research under contract N00014-75-C-0144.

have 'perturbed' input-output relations which may in general include unmodeled 'cross-couplings' between components as well as uncertainty in the input-output relations of the individual components. Thus, we are considering the class of systems having an overall input-output relation

$$\underline{y} = \underset{\sim}{T}\,\underline{u}$$

where the operator $\underset{\sim}{T}$ is defined by feedback equations of the general form (See Figure 1)

$$\begin{bmatrix} y \\ \\ e \end{bmatrix} = \underset{\sim}{H} \begin{bmatrix} u \\ \\ f \end{bmatrix} \qquad (1a)$$

$$\underline{f} = (\underset{\sim}{C} + \delta\underset{\sim}{C})\,\underline{e} \qquad (1b)$$

where

- $\underset{\sim}{H}$ is an LTI operator (having transfer function matrix $\underline{H}(s)$) representing the internal and external interconnections of the system's components;
- $\underset{\sim}{C} = \mathrm{diag}\,(\underset{\sim}{C}_1, \ldots, \underset{\sim}{C}_N)$; (2)[1]
- $\delta\underset{\sim}{C}$ represents the uncertain perturbations and may in general be nonlinear, dynamical, and time-varying;
- $\underline{e} = \mathrm{col}\,(\underline{e}_1, \ldots, \underline{e}_N)$ and
  $\underline{f} = \mathrm{col}\,(\underline{f}_1, \ldots, \underline{f}_N)$;
  $\underline{e}_i$ and $\underline{f}_i$ represent the input and output signals respectively of the i-th 'perturbed' component;
- $\underline{u}$ is the exogenous input to the system;
- $\underline{y}$ is the observed output of the system.

To keep the mathematics tractable, it is assumed that $\underline{H}(s)$ and $\underline{C}(s)$ are rational transfer function matrices -- this is not a serious restriction since most systems of interest with non-rational transfer functions can be approximated arbitrarily closely with respect to the L_2-norm by systems with rational transfer functions.

Our ignorance regarding the accuracy of the 'approximate' models of the system components is assumed to be bounded by an L_2-conic sector. Conic sectors are defined as follows:

Definition: Given any three compatible operators $\underset{\sim}{C}$, $\underset{\sim}{R}$, $\underset{\sim}{S}$, we define L_2-Cone $(\underset{\sim}{C}, \underset{\sim}{R}, \underset{\sim}{S})$ to be the set of ordered pairs of signals $(\underline{x},\underline{y}) \in L_2 \times L_2$ satisfying

$$\|\underset{\sim}{S}(\underline{y} - \underset{\sim}{C}\,\underline{s})\|_{L_2} \leq \|\underset{\sim}{R}\,\underline{x}\|_{L_2}. \qquad (2)^{2}$$

---

[1] For any collection of operators $\underset{\sim}{A}_i$ $(i = 1,\ldots,n)$, the notation diag($\underset{\sim}{A}_1$,... diag($\underset{\sim}{A}_1,\ldots,\underset{\sim}{A}_n$) is defined by

$$\mathrm{diag}(\underset{\sim}{A}_1,\ldots,\underset{\sim}{A}_n) \begin{bmatrix} \underline{e}_1 \\ \vdots \\ \underline{e}_n \end{bmatrix} \triangleq \begin{bmatrix} \underset{\sim}{A}_1\underline{e}_1 \\ \vdots \\ \underset{\sim}{A}_n\underline{e}_n \end{bmatrix}$$

for all $\underline{e}_i$ $(i = 1,\ldots,n)$.

[2] We use the notation L_2 to denote the normed space of C^n-valued signals $\underline{z} : R \to C^n$ for some integer n for which the norm $\|\underline{z}\|_{L_2}^2 \triangleq \langle \underline{z}, \underline{z}\rangle_{L_2}$ exists and is finite; $\langle \underline{z}_1, \underline{z}_2\rangle = \int_{-\infty}^{\infty} \underline{z}_1^T(t)\,\underline{z}_2(t)\,dt$ [5]; $\underline{z}_1^T$ is the complex congugate of $\underline{z}_1$.

The set L_2-Cone$(\underset{\sim}{C},\underset{\sim}{R},\underset{\sim}{S})$ is called an L_2 conic sector with *center* $\underset{\sim}{C}$, *input radius* $\underset{\sim}{R}$, and *output radius* $\underset{\sim}{S}$. If $\underset{\sim}{H}$ is an operator with the property that $\underset{\sim}{H}-\underset{\sim}{C}$ is L_2-stable[3] and if for all $(\underline{x},\underline{y}) \in L_2 \times L_2$ with $\underline{y} = \underset{\sim}{H}\,\underline{x}$ we have

$$\|\underset{\sim}{S}(\underline{y} - \underset{\sim}{C}\,\underline{x})\|_{L_2}^2 \le \|\underset{\sim}{R}\,\underline{x}\|_{L_2}^2 \qquad (3)$$

then we say

$$\underset{\sim}{H}\ \underline{\text{inside}}\ L_2\text{-Cone}\,(\underset{\sim}{C},\ \underset{\sim}{R},\ \underset{\sim}{S}). \qquad (4)$$

[4] (End of Definition)

It is a trivial consequence of Parseval's theorem (cf. [5, p. 236]) that, in the case of L_2-stable LTI operators $\underset{\sim}{S}$, $\underset{\sim}{R}$, and $\delta\underset{\sim}{C} \triangleq \underset{\sim}{H}-\underset{\sim}{C}$ having respective transfer functions $\underline{S}(s)$, $\underline{R}(s)$ and $\underline{H}(s)-\underline{C}(s)$, the condition (4) is equivalent to the frequency-domain condition

$$\sigma_{MAX}\left(\underline{S}(j\omega)\left(\underline{H}(j\omega) - \underline{C}(j\omega)\right)\underline{R}^{-1}(j\omega)\right) \le 1 \qquad (5)$$

[5]

for all ω

(where $\sigma_{MAX}(\underline{A})$ denotes the largest *singular value* of the complex matrix $\underline{A}$ i.e., the largest eigenvalue of $\underline{A}^*\underline{A}$ (or, equivalently, of $\underline{A}\underline{A}^*$) -- cf. [7, Ch. 1]). The motivation for the designation of $\underset{\sim}{C}$, $\underset{\sim}{R}$, and $\underset{\sim}{S}$ as the *center*, *input radius*, and *output radius* respectively of L_2-Cone$(\underset{\sim}{C},\ \underset{\sim}{R},\ \underset{\sim}{S})$ is that in the special case where $\underset{\sim}{H},\underset{\sim}{C},\underset{\sim}{R}$ and $\underset{\sim}{S}$ are SISO and LTI, then the statement $\underset{\sim}{H}$ inside L_2-Cone $(\underset{\sim}{C},\ \underset{\sim}{R},\ \underset{\sim}{S})$ implies that the ω-dependent function $H(j\omega) : C \to C$ given by $Y(j\omega) = H(j\omega)\ U(j\omega)$ maps inputs $U(j\omega)$ lying inside the circle of radius $R(j\omega)$ in the complex plane $\mathbb{C}$ into outputs $Y(j\omega)$ lying inside the circle of center $C(j\omega)$ and radius $S(j\omega)$ in the complex plane $\mathbb{C}$.

For memoryless operators $(\underset{\sim}{H}\,\underline{x})(t) = \underline{h}(\underline{x}(t),\ t)$, $(\underset{\sim}{C}\,\underline{x})(t) = \underline{C}\,\underline{x}(t)$, $(\underset{\sim}{R}\,\underline{x})(t) = \underline{R}\,\underline{x}(t)$, and $(\underset{\sim}{S}\,\underline{y})(t) = \underline{S}\,\underline{y}(t)$, the condition (4) is equivalent to the condition

$$\left\|\underline{S}\left(\underline{h}(\underline{x},\ t) = \underline{C}\,\underline{x}\right)\right\|_{R^n} \le \|\underline{R}\,\underline{x}\|_{R^n} \qquad (6)$$

(where $\|\underline{z}\|_{R^n} \triangleq \sqrt{\underline{z}^T\underline{z}}$ for all $\underline{z} \in R^n$. The motivation for using the term 'conic sector' comes from the fact that in the special case where S, h(·), C, and R are scalar, condition (6) implies that for each t the graph of h(x, t) vs. x is in a cone shaped subset of the real plane -- cf. [13].

Our results assume that

$$\underset{\sim}{C} + \delta\underset{\sim}{C}\ \underline{\text{inside}}\ L_2\text{-Cone}(\underset{\sim}{C},\ \underset{\sim}{R}_c,\ \underset{\sim}{S}_c) \qquad (7)$$

where $\underset{\sim}{C}$ is as in equation (1) and $(\underset{\sim}{R}_c,\ \underset{\sim}{S}_c)$ are given. It is further assumed that $(\underset{\sim}{R}_c,\ \underset{\sim}{S}_c)$ are L_2-stable LTI operators with square, rational transfer function matrices $(\underline{R}_c(s),\ \underline{S}_c(s))$ having the property that $\underline{R}_c{}^*(j\omega)\underline{R}_c(j\omega)$ and $\underline{S}_c{}^*(j\omega)\underline{S}_c(j\omega)$ are uniformly positive definite for all ω.[6]

---

[3] An operator $\underset{\sim}{H}$ is said to be L_2-stable if $\underset{\sim}{H}\,\underline{x}\ L_2$ whenever $\underline{x}\ L_2$ and further, for some constant $k < \infty$ $\|\underset{\sim}{H}\,\underline{x}\|_{L_2} \le k\,\|\underline{x}\|_{L_2}$ [5].

[4] In the multivariable generalization of the circle stability criterion in [6], a set called L_{2e}-Cone$(\underset{\sim}{C},\ \underset{\sim}{R},\ \underset{\sim}{S})$ is employed which is closely related, but different, from the set L_2-Cone$(C,\ R,\ S)$. In general neither set is a proper subset of the other, but if $\underset{\sim}{H}$, $\underset{\sim}{C}$, $\underset{\sim}{S}$, and $\underset{\sim}{R}^{-1}$ are stable nonanticipative operators, then it can be shown (using the definition in [6] of L_{2e}-Cone$(\cdot,\cdot,\cdot)$) that $\underset{\sim}{H}$ inside L_{2e}-Cone$(\underset{\sim}{C},\underset{\sim}{R},\underset{\sim}{S})$ if and only if $\underset{\sim}{H}$ inside L_2-Cone$(\underset{\sim}{C},\ \underset{\sim}{R},\ \underset{\sim}{S})$.

[5] Of course $\underline{R}^{-1}(j\omega)$ must exist for (5) to be equivalent.

[6] For any matrix $\underline{A}$, the transpose of $\underline{A}$ is denoted $\underline{A}^T$ and the complex congugate of $\underline{A}^T$ is denoted $\underline{A}^*$.

In the special case when there is no ignorance regarding the interconnection structure of the system (i.e., when the perturbation $\delta\underset{\sim}{C}$ contains no 'cross-coupling' between components), then $\delta\underset{\sim}{C}$ takes the special form

$$\delta\underset{\sim}{C} = \mathrm{diag}\ (\delta\underset{\sim}{C}_1, \ldots, \delta\underset{\sim}{C}_N)$$

where $\delta\underset{\sim}{C}_i$ represents the ignorance regarding the dynamics of the i-th component $(i = 1, \ldots, N)$. If there are conic sector bounds available for the uncertainty in the individual components' input-output relations

$$\underset{\sim}{C}_i + \delta\underset{\sim}{C}_i \ \underline{\text{inside}}\ L_2\text{-Cone}\ (\underset{\sim}{C}_i, \underset{\sim}{R}_i, \underset{\sim}{S}_i),\ (i = 1, \ldots, N) \tag{8}$$

then it follows (cf. [9, Lemma 6.2 (vi)], [10, Lemma 4.2 (vi)]) that (5) holds with

$$\underset{\sim}{R}_c = \mathrm{diag}\ (\underset{\sim}{R}_1, \ldots, \underset{\sim}{R}_N) \tag{9}$$

$$\underset{\sim}{S}_c = \mathrm{diag}\ (\underset{\sim}{S}_1, \ldots, \underset{\sim}{S}_N)\ . \tag{10}$$

We denote by $\underset{\sim}{L}$ the LTI operator whose transfer function matrix is (suppressing the argument 's')

$$\underline{L} \equiv \left[\begin{array}{c|c} \underline{L}_{yu} & \underline{L}_{yv} \\ \hline \underline{L}_{eu} & \underline{L}_{ev} \end{array}\right] = \left[\begin{array}{c|c} \underline{H}_{yu} + \underline{H}_{yf}\,\underline{C}(\underline{I} - \underline{H}_{ef}\,\underline{C})^{-1}\underline{H}_{eu} & \underline{H}_{yf}(\underline{I} - \underline{C}\,\underline{H}_{ef})^{-1} \\ \hline (\underline{I} - \underline{H}_{ef}\,\underline{C})^{-1}\,\underline{H}_{eu} & \underline{H}_{ef}(\underline{I} - \underline{C}\,\underline{H}_{ef})^{-1} \end{array}\right] \tag{11}$$

For any full rank rational para-hermitian matrix $\underline{A}(s)$ for which $\underline{A}(j\omega)$ positive definite for all ω, we denote by $A^{1/2}(s)$ the rational spectral factor of $\underline{A}(s)$ (unique to within a constant unitary left multiplier) having the properties that $\underline{A}(s) = (\underline{A}^{1/2}(-s))^T\,\underline{A}^{1/2}(s)$, that $(\underline{A}^{1/2}(s))^{-1}$ exists, and that $\underline{A}^{1/2}(s)$ and its inverse have no poles in $\mathrm{Re}(s) \leq 0$; $\underline{A}^{1/2}(s)$ always exists [8, Theorem 2]. If $\underset{\sim}{A}$ is an operator, not necessarily nonanticipative, whose bilateral Laplace transform transfer function matrix is proper, rational, para-hermitian and has the property that $\underline{A}(j\omega)$ is uniformly positive-definite for all ω, then $\underset{\sim}{A}^{1/2}$ denotes the nonanticipative minimum phase LTI operator having transfer function $\underline{A}^{1/2}(s)$. Given any L_2-stable LTI operator $\underset{\sim}{A}$, we denote by $\underset{\sim}{A}^*$ the L_2 adjoint operator; i.e., denoting by $A_\delta(t)$ the impulse response matrix of $\underset{\sim}{A}$, $\underset{\sim}{A}^*$ is the LTI operator with impulse response matrix $\underline{A}_\delta^T(-t)$. Note that if $\underline{A}(s)$ is the bilateral Laplace transform transfer function matrix of $\underset{\sim}{A}$, then $A^T(-s)$ is the transfer function matrix of $\underset{\sim}{A}^*$ and further $\underline{A}^T(-j\omega) = \underline{A}^*(j\omega)$.

MAIN RESULT

Our main result is the following theorem giving tight bounds on the overall systems input-output relation $\underset{\sim}{T}$.

Theorem 1: Suppose that (7) holds. If

(a) uniformly for all ω

$$\sigma_{\min}\ (\underline{S}_c(j\omega)\ \underline{L}_{ev}^{-1}(j\omega)\ \underline{R}_c^{-1}(j\omega)) > 1 \tag{12}$$

(where $\sigma_{\min}(\underline{A})$ denotes the smallest singular value of $\underline{A}$), and

(b) the system (1) is L_2-stable for all $\delta \underset{\sim}{C}$ satisfying (7),[7]

then, provided $L^{*}_{yv}(j\omega)\, L_{yv}(j\omega)$ is full rank for all ω,

$$\underset{\sim}{T} \ \underline{\text{inside}} \ L_2\text{-Cone}\,(\underset{\sim}{T}_{nom},\ \underset{\sim}{P}_T^{1/2},\ \underset{\sim}{Q}_T^{1/2}) \tag{13}$$

where $\underset{\sim}{T}_{nom}$, $\underset{\sim}{Q}_T$, and $\underset{\sim}{P}_T$ are the (not necessarily causal) L_2-stable LTI operators specified in terms of their bilateral Laplace transform transfer matrices by

$$\underline{T}_{nom}(s) \triangleq \underline{L}_{yu}(s) + \underline{L}_{yv}(s)\Big(\underline{S}_c^T(-s)\underline{S}_c(s) - \underline{L}_{ev}^T(-s)\underline{R}_c^T(-s)\underline{R}_c(s)\underline{L}_{ev}(s)\Big)^{-1} \cdot \underline{L}_{ev}^T(-s)\underline{R}_c^T(-s)\underline{R}_c(s)\underline{L}_{eu}(s) \tag{14}$$

$$\underline{Q}_T(s) \triangleq \Big(\underline{L}_{yv}(s)\ \underline{S}_c^T(-s)\underline{S}_c(s) - \underline{L}_{ev}^T(-s)\underline{R}_c^T(-s)\underline{R}_c(s)\underline{L}_{ev}(s)\ ^{-1}\underline{L}_{yv}^T(-s)\Big)^{-1} \tag{15}$$

$$\underline{P}_T(s) \triangleq \underline{L}_{eu}^T(-s)\underline{R}_c^T(-s)\Big(\underline{I} - \underline{R}_c(s)\underline{L}_{ev}(s)\ \underline{S}_c^T(-s)\underline{S}_c(s)\ ^{-1}\underline{L}_{ev}(-s)\underline{R}_c^T(-s)\Big)^{-1} \cdot \underline{R}_c(s)\underline{L}_{eu}(s). \tag{16}$$

Moreover, the set L_2-Cone $(\underset{\sim}{T}_{nom},\ \underset{\sim}{P}_T^{1/2},\ \underset{\sim}{Q}_T^{1/2})$ is the tightest possible bound on $\underset{\sim}{T}$ in the sense that for every pair of signals $(\underline{u}, \underline{y}) \in L_2$-Cone $(\underset{\sim}{T}_{nom},\ \underset{\sim}{P}_T^{1/2},\ \underset{\sim}{Q}_T^{1/2})$, there exists at least one 'perturbation' $\delta \underset{\sim}{C}$ satisfying (7) such that $\underline{y} = \underset{\sim}{T}\,\underline{u}$.

Proof. See Appendix .

DISCUSSION

The uncertainty bound provided by Theorem 1, like the uncertainty bound on the plant ignorance $\delta \underset{\sim}{C}$, takes the form of an L_2-conic sector condition. Consequently, in the special case in which the perturbation $\delta \underset{\sim}{C}$ is linear-time-invariant, Parseval's theorem [5, p. 236] implies that the L_2-conic sector bound on the overall system input-output relation provided by Theorem 1 can be expressed using singular values giving the bound on the system's frequency response matrix

$$\sigma_{max}\big(Q_T^{1/2}(j\omega)\,(T(j\omega) - T_{nom}(j\omega))\ P_T^{-1/2}(j\omega)\big) \le 1\,,$$

cf. equation (6). In particular for single-input-single-output LTI systems having uncertain LTI components, this bound becomes

$$|T(j\omega) - T_{nom}(j\omega)|^2 \le \frac{P_T(j\omega)}{Q_T(j\omega)} \quad .$$

---

[7] For nonanticipative systems, L_2-stability is implied by L_{2e}-stability (which is defined in [6]). It can be shown that if the system (1) is nonanticipative and is L_{2e}-stable when $\delta \underset{\sim}{C} \equiv 0$, then (12) is sufficient to guarantee that (1) is L_{2e}-stable for all $\delta \underset{\sim}{C}$ satisfying (7) [6]. Consequently, at least for nonanticipative systems, condition (b) of Theorem 1 is not very restrictive and may be easily verified by checking that $\underset{\sim}{L}$ is L_{2e}-stable.

Theorem 1 gives the tightest possible bound on the overall system input-output relation T only when all available information about the perturbation $\delta \underset{\sim}{C}$ is contained in condition (5). This is an important limitation, since in general this will not be the case. For example, if bounds on system uncertainty are given in the form of (8) then, although condition (7) holds with $\underset{\sim}{R}_C$ and $\underset{\sim}{S}_C$ specified by (9)-(10), condition (7) does not retain the 'structural' information that $\delta\underset{\sim}{C}$ is diagonal (i.e., $\delta \underset{\sim}{C}$ has no 'cross-couplings' between components). Consequently, in such situations, one may in general, expect the bound provided by Theorem 1 to be conservative. In contrast, the classical root-locus and frequency-response methods (cf. [3, Ch. 6] and [4]) for generating 'regions of uncertainty' (viz., root loci and frequency response 'templates') to bound system response need not be conservative. Thus, the response bounds provided by Theorem 1 are the most useful in the case of systems for which these classical approaches are impractical or inapplicable. Such systems include multiloop feedback systems, multi-input-multi-output systems, systems with more than a very few uncertain parameters, and systems with nonlinear components.

The problem formulation used in this paper is sufficiently general to accommodate multiloop 'two degree of freedom' (cf. [12, §6.1], [4]) feedback structures with plant parameter ignorance such as depicted in Figure 2. For this configuration, one finds that the transfer function matrix $\underline{L}(s)$ of (11) is given by (suppressing the argument 's')

$$\left[\begin{array}{c|c} \underline{L}_{yu} & \underline{L}_{yv} \\ \hline \underline{L}_{eu} & \underline{L}_{ev} \end{array}\right] = \left[\begin{array}{c|c} \underline{G}_0\,\underline{C}(\underline{I} + \underline{F}\,\underline{C})^{-1} & \underline{G}_0(\underline{I} + \underline{C}\,\underline{F})^{-1} \\ \hline (\underline{I} + \underline{F}\,\underline{C})^{-1}\,\underline{P} & -\underline{F}(\underline{I} + \underline{C}\,\underline{F})^{-1} \end{array}\right]$$

where $\underline{F} = \underline{G}_F + \underline{G}_I\,\underline{G}_C\,\underline{G}_{FC}$ and $P = \underline{G}_{IC}\,\underline{G}_C\,\underline{G}_{PC}$. As one might expect based on the work of Horowitz et al.(e.g., [4], [12]), it can be shown using Theorem 1 that the 'two degree of freedom' structure of Figure 2 permits one to adjust the amount of system uncertainty and the nominal value $\underset{\sim}{T}_{nom}$ of the closed-loop system input-output relation independently to a certain extent -- this is the subject of a forthcoming paper.

ACKNOWLEDGEMENT

I would like to thank all those whose criticisms and comments contributed to this paper. In particular, I thank Mr. J. Doyle of Honeywell, Inc. and Mr. P. Van Dooren of the University of Southern California.

APPENDIX

Proof of Theorem 1

We begin by establishing a useful lemma.

Lemma A1: If $\underset{\sim}{Q} \in \mathbb{C}^{n\times n}$ is a positive definite hermitian matrix and if $\underline{B} \in \mathbb{C}^{r\times n}$ is a matrix of full rank $r \leq n$, then the matrix $\underline{Q} = (\underline{B}\,\underline{Q}^{-1}\underline{B}^*)^{-1}$ exists and has the following properties:

i) $\underline{\tilde{Q}} - \underline{B}^*\underline{\tilde{Q}}\,\underline{B}$ is positive semi definite;

ii) if $\underline{S}$ is any matrix such that $\underline{Q} - \underline{B}^*\underline{S}\,\underline{B}$ is positive semi definite, then $\underline{\tilde{Q}} - \underline{S}$ is positive semidefinite

iii) $$(\underline{B}^+)^*\underline{Q}\,\underline{B}^+ = (\underline{B}^+)^*\underline{B}^*\underline{\tilde{Q}}\,\underline{B}\,\underline{B}^+ \qquad (A1)$$

where

$$\underline{B}^{+} \triangleq \underline{Q}^{-1}\underline{B}^{*}\tilde{\underline{Q}} \quad . \tag{A2}$$

Proof: For any positive definite matrix M and any positive integer m, let C_M^m denote the inner product space of complex m-vectors with inner product $\langle \underline{z}_1, \underline{z}_2 \rangle \triangleq \underline{z}_1^{*}\,\underline{M}\,\underline{z}_2$. Consider $\underline{B}$ as a linear mapping C_Q^n into C_I^r. Then the adjoint of $\underline{B}$, which we denote as $\underline{B}^a$, is

$$\underline{B}^{a} = \underline{Q}^{-1}\,\underline{B}^{*}$$

(recall $\underline{B}^{*}$ denotes the complex-conjugate of $\underline{B}^{T}$) and $\underline{B}^{+}$ is the pseudo-inverse of $\underline{B}$ [11, pp. 150-165]. Further,

$$\begin{aligned}\tilde{\underline{Q}} \triangleq (\underline{B}\,\underline{Q}^{-1}\,\underline{B}^{*})^{-1} &= ((\underline{B}\,\underline{Q}^{-1}\underline{B}^{*})^{-1}\underline{B}\,\underline{Q}^{-1})\underline{Q}(\underline{Q}^{-1}\underline{B}^{*}(\underline{B}\,\underline{Q}^{-1}\underline{B}^{*})^{-1})\\ &= (\underline{B}^{+})^{*}\,\underline{Q}\,\underline{B}^{+}.\end{aligned} \tag{A3}$$

For any vector $\underline{x} \in C_Q^n$, let $\underline{x}_{\eta(B)}$ and $\underline{x}_{\eta^{\perp}(B)}$ denote the respective projections of $\underline{x}$ onto the nullspace of $\underline{B}$ and the orthogonal complement of the nullspace of $\underline{B}$. Now, for any $\underline{x} \in C_Q^n$ and any $\underline{y} \in C_I^r$

$$\underline{B}^{+}\,\underline{B}\,\underline{x} = \underline{x}_{\eta^{\perp}(B)} \quad , \tag{A4}$$

$$\underline{B}\,\underline{B}^{+}\,\underline{y} = \underline{y} \quad . \tag{A5}$$

It follows that for all $\underline{x}$

$$\underline{x}^{*}\;(\underline{Q} - \underline{B}^{*}\,\tilde{\underline{Q}}\,\underline{B})\;\underline{x} = \underline{x}_{\eta(B)}^{*}\;\underline{Q}\,\underline{x}_{\eta(B)} \geq 0$$

which establishes property (i). For any matrix S satisfying $\underline{Q} - \underline{B}^{*}\underline{S}\,\underline{B} \geq 0$ and any $\underline{y} \in C_I^r$ we have

$$\begin{aligned}\underline{y}^{*}\;\underline{S}\;\underline{y} &= \underline{y}^{*}\;(\underline{B}\underline{B}^{+})^{*}\;\underline{S}\;(\underline{B}\,\underline{B}^{+})\;\underline{y}\\ &= (\underline{B}^{+}\,\underline{y})^{*}\;\underline{B}^{*}\;\underline{S}\,\underline{B}\;(\underline{B}^{+}\,\underline{y})\\ &\leq (\underline{B}^{+}\,\underline{y})^{*}\;\underline{Q}\;(\underline{B}^{+}\,\underline{y})\\ &= \underline{y}^{*}\,\tilde{\underline{Q}}\,\underline{y} \quad ,\end{aligned} \tag{A6}$$

which establishes property (ii). The identity of property (iii) follows by direct substitution of (A2) into (A1).

▭

We now prove that, under the conditions of Theorem 1, equation (13) holds. We begin by noting the input-output relation $\underline{y} = \underset{\sim}{T}\,\underline{u}$ defined by (1) can be described equivalently by

$$\begin{bmatrix}\underline{y}\\ \underline{e}\end{bmatrix} = \underset{\sim}{L}\begin{bmatrix}\underline{u}\\ \underline{v}\end{bmatrix} = \begin{bmatrix}(\underset{\sim}{L}_{yu}\,\underline{u} + \underset{\sim}{L}_{yv}\,\underline{v})\\ (\underset{\sim}{L}_{eu}\,\underline{u} + \underset{\sim}{L}_{ev}\,\underline{v})\end{bmatrix} \tag{A7}$$

$$\underline{v} \triangleq \delta\,\underset{\sim}{C}\,\underline{e} \quad ; \tag{A8}$$

we take ($\underline{u}$, $\underline{y}$, e, $\underline{f}$, $\underline{v}$) to be a solution of (1) and (A7). For any $\underline{z} \in L_2$ and any L_2-stable operator $\underset{\sim}{A}$, we use the notation $\hat{z}$ and $\hat{A}$ to denote the Fourier transform of $\underline{z}$ and frequency response of $\underset{\sim}{A}$ respectively.

Suppose that (7) holds. Then $(\underline{e}, \underline{f}) \in L_2$-Cone $(\underset{\sim}{C}, \underset{\sim}{R}_c, \underset{\sim}{S}_c)$ and hence

$$\begin{aligned}
0 &\geq \|\underset{\sim}{S}_c(\underline{f} - \underset{\sim}{C}\,\underline{e})\|_{L_2}^2 - \|\underset{\sim}{R}_c\,\underline{e}\|_{L_2}^2 \\
&= \langle \underset{\sim}{S}_c\,\underline{f} - (\underset{\sim}{S}_c\underset{\sim}{C} + \underset{\sim}{R}_c)\,\underline{e},\ \underset{\sim}{S}_c\,\underline{f} - (\underset{\sim}{S}_c\underset{\sim}{C} - \underset{\sim}{R}_c)\,\underline{e} \rangle_{L_2} \\
&= \langle (\underset{\sim}{S}_c + \underset{\sim}{R}_c\underset{\sim}{L}_{ev})\underline{v} + \underset{\sim}{R}_c\underset{\sim}{L}_{eu}\underline{u},\ (\underset{\sim}{S}_c - \underset{\sim}{R}_c\underset{\sim}{L}_{ev})v - \underset{\sim}{R}_c\underset{\sim}{L}_{eu}\underline{u} \rangle_{L_2}
\end{aligned}$$

(By (1) and (A7))

$$\begin{aligned}
&= \langle (\hat{S}_c + \hat{R}_c\hat{L}_{ev})\hat{v} + \hat{R}_c\hat{L}_{eu}\,\hat{u},\ (\hat{S}_c - \hat{R}_c\hat{L}_{ev})\,\hat{v} \\
&\qquad - \hat{R}_c\hat{L}_{eu}\,\hat{u} \rangle_{L_2} \\
&= \langle (\hat{v} - \hat{T}_v\,\hat{u}),\ \hat{Q}_v\,(\hat{v} - \hat{T}_v\,\hat{u}) \rangle_{L_2} - \langle \hat{u}, \hat{P}_v\,\hat{u} \rangle_{L_2} \qquad \text{(A9)}
\end{aligned}$$

where $\underset{\sim}{Q}_v$, $\underset{\sim}{Z}$, $\underset{\sim}{\Theta}$, $\underset{\sim}{T}_v$, and $\underset{\sim}{P}_v$ are the (not necessarily causal) L_2-stable operators specified in terms of their bilateral Laplace transform transfer functions by

$$\underline{Q}_v(s) \triangleq \underline{S}_c^T(-s)\,\underline{S}_c(s) - \underline{L}_{ev}^T(-s)\underline{R}_c^T(-s)\underline{R}_c(s)\underline{L}_{ev}(s) \qquad \text{(A10)}$$

$$Z(s) \triangleq L_{ev}^T(-s)\,R_c^T(-s)\,R_c(s)\,L_{eu}(s) \qquad \text{(A11)}$$

$$\Theta(s) \triangleq L_{eu}^T(-s)\,R_c^T(-s)\,R_c(s)\,L_{eu}(s) \qquad \text{(A12)}$$

$$T_v(s) \triangleq Q_v^{-1}(s)\,Z(s) \qquad \text{(A13)}$$

$$P_v(s) \triangleq \Theta(s) + Z^*(s)\,Q_v^{-1}(s)\,Z(s) = P_T(s) \qquad \text{(A14)}$$

where the latter equality in (A14) follows using the matrix identity

$$I + A\,(B - A^*A)^{-1}A^* = (I - A\,B^{-1}A^*)^{-1}. \qquad \text{(A15)}$$

Thus, applying part (i) of Lemma A1, we have from (A9) that

$$\begin{aligned}
0 &\geq \langle \hat{L}_{yv}\,(\hat{v} - \hat{T}_v\,\hat{u}),\ (\hat{L}_{yv}\,\hat{Q}_v^{-1}\,\hat{L}_{yv}^*)^{-1} \\
&\qquad \hat{L}_{yv}\,(\hat{v} - \hat{T}_v\,\hat{u}) \rangle_{L_2} - \langle \hat{u}, \hat{P}_T\,\hat{u} \rangle_{L_2} \\
&= \langle \hat{y} - \hat{T}_{nom}\,\hat{u},\ \hat{Q}_T\,(\hat{y} - \hat{T}_{nom}\,\hat{u}) \rangle_{L_2} \\
&\qquad - \langle \hat{u}, \hat{P}_T\,\hat{u} \rangle_{L_2}. \qquad \text{(A16)}
\end{aligned}$$

Hence, from Parseval's theorem

$$\|\underset{\sim}{Q}_T^{1/2}\,(\underline{y} - \underset{\sim}{T}_{nom}\,\underline{u})\|_{L_2} \leq \|P_T^{1/2}\,\underline{u}\|_{L_2} \qquad \text{(A17)}$$

from which (13) follows.

It remains to prove that whenever $(\underline{u}, \underline{y}) \in L_2$-Cone $(\underset{\sim}{T}_{nom}, \underset{\sim}{P}_T^{1/2}, \underset{\sim}{Q}_T^{1/2})$ there exists a $\underset{\sim}{C}$ satisfying (7) such that $\underline{y} = \underset{\sim}{T}\,\underline{u}$. To see this, let $\underline{v}$ be specified by

$$\hat{v} = \hat{L}_{yv}^{+} (\hat{y} - \hat{L}_{yu}\, \hat{u}) + \hat{T}_v\, \hat{u} \tag{A18}$$

where

$$\hat{L}_{yv}^{+} = \hat{Q}_v^{-1}\, \hat{L}_{yv}^{*} (\hat{L}_{yv}\, \hat{Q}_v^{-1}\, \hat{L}_{yv}^{*})^{-1} \quad . \tag{A19}$$

Then, (u, y, v) are consistant with (1) and (A7) and (using part (iii) of Lemma (A1) and Parseval's theorem) it follows that for all $(\underline{u}, \underline{y}) \in L_2$-Cone $(\underset{\sim}{T}_{nom}, \underset{\sim}{P}_T^{1/2}, \underset{\sim}{Q}_T^{1/2})$

$$\begin{aligned} 0 &\geq \| \hat{Q}_T^{1/2}\, \hat{y} - \hat{T}_{nom}\, \hat{u} \|_{L_2}^2 - \| \hat{P}_T^{1/2}\, \hat{u} \|_{L_2} \\ &= < (\hat{v} - \hat{T}_v\, \hat{u}), \hat{L}_{yv}^{*} (\hat{L}_{yv}\, \hat{Q}_v^{-1}\, \hat{L}_{yv}^{*})^{-1}\, \hat{L}_{yv} (\hat{v} - \hat{T}_v\, \hat{u}) >_{L_2} \\ &\qquad - < \hat{u}, \hat{P}_T\, \hat{u} >_{L_2} \\ &= \| \hat{S}_c\, (\hat{f} - \hat{C}_e) \|_{L_2}^2 - \| \hat{R}_c\, \hat{e} \|_{L_2}^2 \end{aligned} \tag{A20}$$

where the latter equality follows as in (A9). Let $\delta \underset{\sim}{C}$ be the map defined by

$$\delta \underset{\sim}{C}\, \underline{z} = \| \underset{\sim}{R}_c\, \underline{e} \|_{L_2}^{-2} < \underset{\sim}{R}_c\, \underline{e}, \underset{\sim}{R}_c\, \underline{z} >_{L_2} \underline{v}. \tag{A21}$$

Using (A21) and the Schwartz inequality, one readily verifies that $\delta \underset{\sim}{C}\, \underline{e} = \underline{v}$ and that $\delta \underset{\sim}{C}$ satisfies (7).

(End of proof)

REFERENCES

[1] W. R. Evans, "Graphical Analysis of Control Systems," Trans. AIEE, Vol. 67, pp. 547-551, 1948.

[2] H. M. James, N. B. Nichols, and R. S. Phillips, Theory of Servomechanisms. New York: McGraw-Hill, 1947.

[3] R. C. Dorf, Modern Control Systems. Reading, Massachusetts: Addison-Wesley, 1973.

[4] I. M. Horowitz and M. Sidi, "Synthesis of Feedback Systems with Large Plant Ignorance for Prescribed Time-Domain Tolerances," Int. J. Control, Vol. 16, pp. 287-309, 1972.

[5] C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties. New York: Academic Press, 1975.

[6] M. G. Safonov and M. Athans, "A Multiloop Generalization of the Circle Stability Criterion," to appear in Proc. Twelfth Annual Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, California, November 6-8, 1978.

[7] G. Forsythe and C. B. Moler, Computer Solution of Linear Algebraic Systems. Englewood Cliffs, New Jersey: Prentice-Hall, 1967.

[8] D. C. Youla, "On the Factorization of Rational Transfer Function Matrices," IRE Trans. on Information Theory, pp. 172-182, July 1961.

[9] M. G. Safonov and M. Athans, "On Stability Theory," to appear in Proc. IEEE Conf. on Decision and Control, San Diego, California, January 10-12, 1979.

[10] M. G. Safonov, Stability and Robustness of Multivariable Feedback Systems. Cambridge, Massachusetts: The MIT Press, to appear 1979.

[11] D. G. Luenberger, Optimization by Vector Space Methods, New York: Wiley, 1969.

[12] I. M. Horowitz, Synthesis of Feedback Systems. New York: Academic Press, 1963.

[13] G. Zames, "On the Input-Output Stability of Time-Varying Nonlinear Feedback Systems -- Part II: Conditions Involving Circles in the Frequency Plane and Sector Nonlinearities," IEEE Trans. on Automatic Control, Vol. AC-11, pp. 465-576, July 1966.

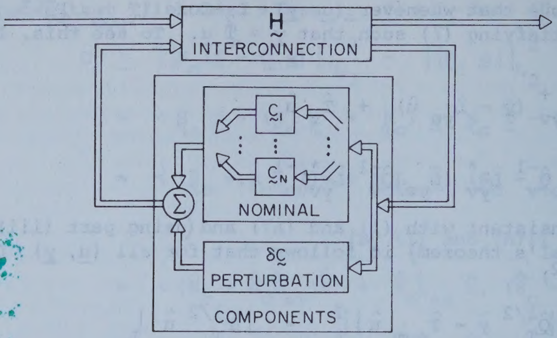

H
INTERCONNECTION
C1
CN
Σ
NOMINAL
δC
PERTURBATION
COMPONENTS

Figure 1. The system.

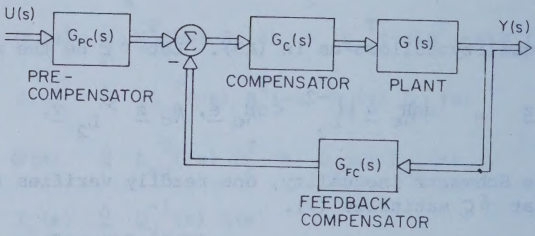

U(s)
GPC(s)
PRE-
COMPENSATOR
Σ
−
GC(s)
COMPENSATOR
G (s)
PLANT
Y(s)
GFC(s)
FEEDBACK
COMPENSATOR

Figure 2(a). Multiloop 'two degree of freedom' feedback control system.

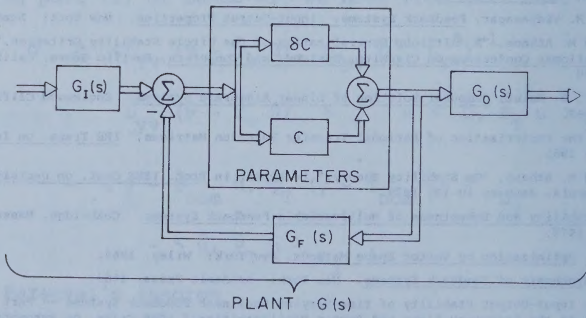

GI(s)
Σ
−
δC
C
Σ
PARAMETERS
GO(s)
GF (s)
PLANT G(s)

Figure 2(b). Internal structure of uncertain plant in Figure 2(a).

SINGULAR VALUES AND FEEDBACK: DESIGN EXAMPLES*

G. STEIN
Honeywell SRC
Minneapolis, MN, and
Massachusetts Institute of Technology
Cambridge, MA

J.C. DOYLE, Consultant
Honeywell SRC
Minneapolis, MN, and
University of California
Berkeley, CA

ABSTRACT

Some control law design examples for the CH-47 helicopter are used to explore and illustrate the role of singular value analyses in multivariable design.

1.0 INTRODUCTION

Design techniques for linear multivariable control systems have long suffered from the lack of reliable measures of "robustness". By robustness, we mean an intentionally designed tolerance for differences between the nominal plant model used for design and the actual plant being controlled. Such differences arise as a result of parameter variations, neglected dynamics, approximated functional relationships, nonlinearities, etc. They are present to some extent in all physical systems. Critical closed loop properties such as stability must therefore be designed to remain intact in the face of these differences, and key performance variables should exhibit only weak dependence on them.

For single input single output systems, the degree to which such tolerance is achieved has been historically (and reliably) measured in terms of the minimum distance of a loop's Nyquist diagram from the so-called critical point (-1,0) in the complex plane [1]. The familiar concepts of phase margin and gain margin are measures of this distance and are often specified as explicit minimum robustness requirements of control loops [for example, 2].

Many attempts have been made to apply these measures of robustness to multivariable systems. A common engineering practice, for example, is to measure gain and phase margins of individual loops one at a time with other loops variously open or closed. More formal methods have been advocated by Rosenbrock [3], who applies these measures to the inverse Nyquist plots of diagonally dominant systems, and by MacFarlane et al. [4], who apply them to the eigenvalue (or characteristic loci) plots of multivariable systems. All of these approaches have been shown to be unrealistic in the sense that a lack of tolerance in certain directions goes undetected [5]. In effect, the methods can indicate that all is well with respect to robustness when dangerous sensitivities in fact exist.

The concept of singular values of matrix transfer functions has recently been applied to overcome this reliability problem. In [5], Doyle established a multivariable stability-robustness theorem which guarantees that a stable multivariable control system will remain stable in the face of multiplicative model perturbations whenever the singular values of these perturbations remain appropriately bounded. This result has since been shown by Sandell [6] to be a special case of a general invertability condition for stable perturbed operators. It is also implicit in the recent stability results of Safonov [7]. When it is combined with

---

*This work was supported by the Office of Naval Research under Contract No. N00014-75-C-0144.

efficient computational procedures for singular value decomposition and with tight bounding formulas for the magnitudes of model perturbations the result appears to provide a valuable new multivariable design tool.

It is the intent of this paper to explore and illustrate the role which singular value analyses might play in multivariable design. We do this by means of some trial control design examples for the longitudinal degrees of freedom of the CH-47 helicopter. In forward flight, this vehicle exhibits coupled pitch attitude and vertical motion dynamics which must be controlled by coordinated actuation of two inputs. This offers a realistic, yet manageable design example. We begin the discussions in Section II with a quick review of Doyle's stability-robustness result and of some bounding formulas for model perturbations recently developed by Safonov [8]. Specifics of the CH-47 design problem are given in Section III and IV, and various trial designs are discussed in Section V. Conclusions and some open research questions are given in Section VI. We caution all readers to consider the control laws presented as illustrative only. They are not, and are not intended to be, final "flight quality" designs.

2.0 MULTIVARIABLE ROBUSTNESS CONCEPTS

We consider finite dimensional, linear, time-invariant multivariable feedback loops represented in the frequency domain as shown in Figure 1*. The matrix $G_0(s)$ represents the nominal plant transfer function, and $L(s)$ represents a multiplicative perturbation such that the actual plant is given by

$$G(s) = G_0(s)\,[I+L(s)]. \tag{1}$$

If $L(s)$ and the nominal closed loop system, $(I+G_0)^{-1}G_0$, are both stable then the following result holds:

Theorem

The perturbed closed loop system remains stable for all perturbations $L(s)$ such that

$$\bar{\sigma}[L(s)] < \underline{\sigma}[I+G_0^{-1}(s)] \tag{2}$$

for all complex frequencies, s, on the classical Nyquist D-contour.** Here the symbols $\bar{\sigma}[A]$ and $\underline{\sigma}[A]$ denote the maximum and minimum singular values of matrix A, respectively, with singular values being defined as the square roots of eigenvalues of A^*A. Equation (2) immediately suggests that the function $\underline{\sigma}[I+G_0^{-1}(s)]$, $s \epsilon D$, provides a reliable measure of multivariable robustness--the bigger $\underline{\sigma}$, the better. Moreover, since this function is equal for $s=\pm jw$ and tends to infinity on the infinite segment of the D-contour (at least for all physical systems), it suffices to look at the singular values for real positive frequencies only, i.e., the system remains stable if

$$\bar{\sigma}[L(jw)] < \underline{\sigma}[I+G_0^{-1}(jw)] \tag{3}$$

$$\text{for all } 0 \le \omega < \infty.$$

---

*More general systems can be treated with Sandell's [6] and Safonov's [7] generalizations.

**The D-contour encloses the right half plane with three segments: 1) $s=jw$, $0 \le \omega < \infty$; 2) $s=Re^{j\theta}$, $R \to \infty$, $-\frac{\pi}{2} \le \theta \le \frac{\pi}{2}$; 3) $s=jw$ $-\infty < \omega \le 0$. It is usually indented to exclude singularities along this path.

This equation calls for nothing more than "generalized Bode gain plots", (i.e.,$\bar{\sigma}$ and $\underline{\sigma}$ vs.ω) to measure system robustness. We will refer to these plots as sigma-plots for convenience.

In order to utilize equation (3), it is necessary to be able to calculate singular values efficiently and to be able to characterize realistic perturbations L(s) for complex systems. Numerical procedures for the first requirements are discussed by Laub in [9]. The second requirement is satisfied to substantial degree by the following composite sectoricity results due to Safonov [8]*.

Consider the system diagram shown in Figure 2. The matrices G_O, G_{yy}, G_{eu} and G_{ey} are nominal transfer functions of our original plant with its block diagram redrawn such that the uncertainties of individual components are all collected into one perturbation matrix; dia $[c_i(s)\ i = 1,2,\ldots,N]$. The c_i's are nominally zero and are assumed to have known individual bounds

$$|c_i(jw)| \leq |r_i(jw)| \tag{4}$$

where the r_i's are stable, minimum phase, rational transfer functions. Let $R \triangleq \mathrm{diag}(r_i)$ and let

a) the nominal system be stable, and

b) the matrix $M \triangleq I-G^*_{ev} R^* RG_{ev}$ be uniformly positive for all ω.

Then the perturbed system's transfer function matrix will belong to a "conic sector" (a circle in the frequency domain) defined by

$$\bar{\sigma}[Q^{\frac{1}{2}}(jw)(G(jw)-G_c(jw))\ P^{-\frac{1}{2}}(jw)] \leq 1 \text{ for all } \omega$$

with

$$G_c \triangleq G_o + G_{yv}M^{-1}\ G^*_{ev}\ R^*RG_{eu} \tag{6}$$

$$Q \triangleq [G_{yv}\ M^{-1} G^*_{yv}]^{-1} \tag{7}$$

$$P \triangleq G^*_{eu}\ R^*\ [I-RG_{ev} G^*_{ev}\ R^*\ ]^{-1}\ RG_{eu} \tag{8}$$

Note that equations (5)-(8) provide a way to compute bounds for the total perturbation L(s) in Figure 1 from know bounds for individual component elements of the plant.

III. THE EXAMPLE DESIGN PROBLEM

To examine the potential utility of the above concepts in multivariable system design, we will treat a longitudinal-axes design problem for the CH-47 helicopter. This vehicle is a tandem rotor machine whose physical characteristics and mathematical models are given in [10]. Control over vertical motions is achieved by simultaneous changes of blade angle-of-attack on both rotors (collective), while pitch and forward motions are controlled by changing blade angle differentially between the two rotors (differential-collective). These blade angle changes are transformed through rotor dynamics and aerodynamics into hub forces which then move the machine.

Our objectives will be to design a command augmentation control law which achieves tight, non-interacting control of the vertical velocity and pitch attitude responses. A small perturbation linearized aircraft model should prove adequate for this purpose and is available from [10]. The state vector consists of the vehicle's basic rigid body variables $x = (V,\dot{z},q,\theta)$ (forward velocity, vertical velocity, pitch rate, pitch angle). Two integrators are appended to achieve integral control of the primary responses, and controls are the collective and differential-collective inputs described avove, u=(c,dc). Hence, the design model is

$$\dot{x} = Ax + Bu \qquad A,B \text{ in } [10] \tag{9}$$

$$\dot{x}_5 = -\dot{z} + \dot{z}_{cmd} \tag{10}$$

$$\dot{x}_6 = -\theta + \theta_{cmd} \tag{11}$$

The major approximations associated with this model are due to neglected dynamics of the rotors, to neglected nonlinearities in the blade angle actuation hardware, and to variations of the A,B matrices with operating point (flight condition variations). We will treat modelling errors due to these approximations as sources of the perturbation L(s) in Figure 1 and will attempt to make controllers robust with respect to them.

IV. CHARACTERIZATION OF L(s)

In view of the stability-robustness theorem cited earlier, it is not unjustified to consider nominal design models to be incomplete unless they are accompanied by estimates of the function $\bar{\sigma}[L(s)]$. (How else can a designer assure the required degree of robustness?) Such estimates are developed in this section for the model in equations (9)-(11).

Perturbations Due to Rotor Dynamics

Elementary dynamic and aerodynamic analyses of rotating airfoils, hinged at the rotor hub, indicate that lift forces will not be transmitted to the hub instantaneously with collective changes in blade angle-of-attack but will appear only when the cone angle of the rotor has appropriately changed. The dynamics of the latter have been shown to be damped second order oscillations with natural frequency equal to rotor speed and damping determined by somewhat uncertain aerodynamic effects [11]. Hence, rotor dynamics can be crudely represented by second order transfer functions

$$g_R(s,\zeta) = \frac{\omega_R^2}{s^2 + 2\zeta\omega_R s + \omega_R^2} \tag{12}$$

with ω_R=25 and ζ conservatively confined to the range $0.1 \le \zeta \le 1.0$. Because collective and differential-collective inputs both involve coning motions of the rotors, one such transfer function will appear in each control channel. Since these dynamics are neglected in equation (9), it then follows that any perturbated transfer function matrix computed from Table 1 will have the form

$$G = G_o(I+L) = G_o \text{ diag } (g_R), \tag{13}$$

and hence,

$$L = \text{diag } (g_R - 1) \tag{14}$$

$$\bar{\sigma}[L] = \max_{\zeta} \left| \frac{s^2 + 2\zeta\omega_R}{s^2 + 2\zeta\omega_R s + \omega_R} \right| \tag{15}$$

This function was evaluated for a range of s=jw values (with brute force maximization over ζ) and is shown by the solid lines in Figure 3.

Figure 3 also shows an alternate bound for $\bar{\sigma}[L]$ developed from Safonov's formulas, equation (5)-(8). In this case, the rotor dynamics are redrawn as in Figure 2, with nominal damping value, $\zeta_o = 0.55$, plus an internal component perturbation, $\delta\zeta$, bounded by

$$|\delta\zeta| \le |r| \quad \text{with } r=0.45 \tag{16}$$

The transfer functions for Figure 2 turn out to be

$$\begin{bmatrix} G_o & G_{yv} \\ G_{eu} & G_{ev} \end{bmatrix} = \begin{bmatrix} \omega_R^2 & -2\omega_R \\ \omega_R^2 s & -2\omega_R s \end{bmatrix} / (s^2+2\zeta_o\omega_R s+\omega_R^2) \tag{17}$$

and according to (5), the rotor dynamics at each frequency then belong to a circle in the complex plane with center, $\bar{g}_R$, defined by (6) and radius, $\tilde{g}_R$, given by $\sqrt{p/q}$ with scalars p and q defined by (7) and (8). Using this circle in (14) gives the bound

$$\bar{\sigma}\,[L] \leq |\bar{g}_R - 1| + \tilde{g}_R, \tag{18}$$

which is plotted in Figure 3. Note that this "Safonov bound" is slightly more conservative than equation (11) because it admits a larger class of damping perturbations--i.e., $\delta\zeta$ in (13) may itself be a dynamical operator.

Perturbations Due to Rate Limits

In addition to the dynamics of rotors, each control channel of the CH-47 also exhibits various nonlinearities which are neglected in the nominal design model. Of these, the rate limit nonlinearity imposes the greatest dynamic constraint on performance, and we consider bounds only for this one effect.

An approximate model for rate limits on the CH-47 is given by

$$\dot{u} = R_{\ell im}\ \mathrm{SAT}\ [\,94(u_c - u)/R_{\ell im}] \tag{19}$$

where SAT( ) denotes the standard saturation nonlinearity, saturating at $\pm$ 1. Bounds for this model can be developed with Safonov's procedure by treating the SAT element as an uncertain component. For example, if we are prepared to restrict our system to functions $\zeta(t)$ whose L_∞-norms are less than some multiple of the limit, say

$$||\zeta||_{L_\infty} \leq \eta, \qquad \eta \geq 1.0, \tag{20}$$

then

$$\mathrm{SAT}(\zeta) = (0.5 + 0.5/\eta) + (\delta c)\zeta \tag{21}$$

$$\text{with } |\delta c| \leq |r| = 0.5 - 0.5/\eta$$

The δc perturbations are nonlinear, of course, and equation (5)-(8) do not apply to them as stated. Fortunately, reference [8] shows that the bounds are still valid for these and other more general perturbations. Hence, the rate limit nonlinearities belong to a conic sector and have a $\bar{\sigma}[L]$ bound analogous to (19). This bound is plotted in Figure 3 (dashed lines) for several values of the magnitude ratio η. Note that as η becomes large, the bound approaches unity at all frequencies. This is consistant with physical intuition, since the effective gain across rate limited nonlinearities will approach zero for large signal levels.

Perturbations Due to Operating Point

The third major source of model uncertainty is the variation of A, B matrices with flight condition. Such "component" variation could again be translated into an overall bound for L(s) via Safonov's procedure. In this case, however, the result would be unduly conservative because coefficient variations tend to be highly correlated and are not arbitrary dynamical operators. A more direct way to compute the bound is to solve (1) for L(s) with known G(s) matrices and to maximize over a number of representative flight conditions, i.e.,

$$\bar{\sigma} L(jw) = \max_{i} \bar{\sigma} [G_0^{-1}(jw)\, G_i(jw) - I] \qquad (22)$$

Results of this process are shown by the dotted line in Figure 3. We see the (initially surprising result that $\bar{\sigma}[L]$ becomes quite large at low frequencies. This happens because the basic helicopter's low frequency modes are stable at some flight conditions and unstable at others. (Theoretically, $\bar{\sigma}$ G(jw) will approach infinity for frequencies and flight conditions where these modes cross the jw-axis. This means that the perturbations exhibited by our plant are not necessarily stable and, hence, the stability-robustness theorem cited in Section II fails to apply. We will see later that stable controllers can still be obtained and that the ability to incorporate unstable L's in a generalized multivariable stability-robustness theory appears to be an important research topic. For the moment, however, our designs will be restricted to individual flight conditions for which the dotted L's in Figure 3 can be disregarded.

V. TRIAL CONTROLLER DESIGNS

A bit of contemplation of equation (3) and the uncertainties given in Figure 3 shows that the stability-robustness theorem basically works by imposing a "multivariable bandwidth" limitation on the feedback loop. Magnitudes of L(jw) tend to be large (unity or greater) beyond certain frequencies, requiring G_0^{-1} to be large and consequently G_0 to be small. This is most readily illustrated with a single loop example where plots of the function $\underline{\sigma}[1+g^{-1}]$ reduce to the inverse closed loop frequency response, i.e.,

$$\underline{\sigma}[1+g^{-1}] = \left|\frac{1+g}{g}\right| = \left|\frac{1}{g_{cL}}\right| . \qquad (23)$$

The condition that $\underline{\sigma}[1+g^{-1}]$ be large then translates directly into the high-frequency "roll-off" requirement commonly imposed on classical control loops [1]. We begin with such an example.

Single Loop Pitch Attitude Control

The vertical velocity and pitch attitude motions of the nominal CH-47 model at hover uncouple naturally into two non-interacting channels-- $(\dot{z},x_5)$ controlled by (c), and (v,q,θ,x_6) controlled by (dc). The hover flight condition thus offers an attractive single loop design case. Sigma-plots for several trail pitch-motion controllers for this case are shown in Figure 4. These controllers were all designed with the linear-quadratic methodology (selected largely for convenience) and correspond to the following cost functional:

$$J = \int_0^\infty (57.3\, x_6)^2 + \rho(dc)^2 \quad dt \qquad (24)^*$$

with ρ = 900., 9.0, 0.09, and 1.0, respectively, for the four trials. As expected, bandwidth of these controllers increases with decreasing ρ and eventually violates the stability-robustness constraint imposed by neglected rotor dynamics (for the moment we ignore rate limits and flight condition variations). That this violation actually produces instabilities was verified by computing closed loop roots of the trail controllers in the presence of the rotor. Trial 3 is unstable! Our options are therefore to restrict bandwidth to approximately Trial 2 or to provide additional roll-off beyond the maximum 20 db/decade attenuation inherent in LQ-designs [13]. The latter option is illustrated by Trial 4 which uses a ρ-value somewhat smaller than Trial 2 but includes a low-pass filter at

---

*These weighting selections are motivated by the "asymptotic" procedure of Reference [12]. They achieve a single slow mode near the origin for forward speed and a third order (asymptotic) Butterworth pattern for the remaining states.

ω= 12 rad/sec to help avoid the rotor resonance peak. Note that the closed loop frequency responses* are well-shaped for all pure LQ-trials and that Trial 4 achieves extra bandwidth at the expense of slightly larger M-peaks.

Multi-Loop Designs

Maximizing Bandwidth--The beauty of singular values is that the above stability-robustness analyses carry over without change to multivariable systems. This is illustrated in Figure 5 with some trial two-channel designs at a 40 knot forward speed flight condition. These controllers are again of the LQ-type, this time using the cost function,

$$J = \int_0^\infty (x_5)^2 + (57.3\ x_6)^2 + \rho_1(c)^2 + \rho_2(dc)^2 \ dt, \qquad (25)$$

with (ρ_1, ρ_2) = (10000, 900), (9.0, 9.0), and (1.0, 1.0) for the three trials shown. The distinction between Figures 4 and 5 is that Figure 5 shows two sigma-plots for each trial, corresponding to the two singular values of $(I+G^{-1})$. For stability-robustness, the smaller of these values must fall above the sigma-plot of L at all frequencies. The larger value is unspecified. However, in order to maximize bandwidth "in all directions", it is reasonable to adjust the relative weights (ρ_1, ρ_2) such that the two singular values are approximately equal and then to push them jointly to as high a bandwidth as the $\bar{\sigma}[L]$plot permits. (For the moment, we again use only neglected rotor dynamics for L.) This design philosophy is incorporated in the three trials of Figure 5. The first trail has low bandwidth and substantial differences between the two singular values. These differences are reduced and bandwidth is increased in the next trial. The third trial serves to maximize bandwidth by using additional roll-off filters in each control channel.

Transient Response--As seen from these trials, singular value analyses appear to offer a convenient way to maximize multivariable bandwidth subject to stability-robustness limitations. The next design step is to achieve reasonable command responses from the resulting feedback loop. One way to do this is to place a command shaping filter ahead of the loop, as indicated by the dashed box in Figure 1. For feedback loops with integral control on the primary responses, such sophistication is often unnecessary because commands inserted at the integrators (as shown in equations (10) and (11)) produce good transients. This is the case here, as evidenced by the responses of Trail 3 to step attitude and step velocity commands shown in Figure 6. Note that the loops are tight, well damped, and non-interacting as desired.

Rate Limiting--So far we have ignored model uncertainties due to rate limits. This was done because there is no a priori way to select the parameter η for Figure 3, which is determined by the maximum magnitudes of signals in the closed loop. Clearly, for η sufficiently large all our trial designs would violate the resulting $\bar{\sigma}[L]$ bound. That such violations actually correspond to instabilities was verified by repeating the transient responses for Trail 3 with progressively larger attitude commands. Unstable behavior occurs for $\theta_{cmd} \geq$ 18. degrees, with $\eta \simeq$ 60.

In order to improve robustness with respect to rate limits, the following iterative procedure may be used:

1) Assume a signal level limit $\eta \leq \eta_0$
2) Design $I + G_0^{-1}$ consistent with the resulting $\bar{\sigma}[L]$

---

*According to (23), these are given by the sigma-plots of Figure 4 viewed "upside down".

3) Evaluate the actual maximum signal level, η_1, by computing transient responses with worst case commands and/or initial conditions

4) If η_1 and η_0 are substantially different, return to step 1 with $\eta_0 = \eta_0 + \varepsilon(\eta_1 - \eta_0)$.* Stop otherwise

An illustration of the first iteration of this procedure is given in Figure 5 where the assumed signal level $\eta_0 = 20$. (the dashed $\bar{\sigma}[L]$ curve) yields a controller (Trial 2) whose actual signal level is $\eta_1 \leq 0.6$. The associated transient responses are slow but stable. To fine tune this design, a second iteration might be taken with $\eta_0 = 5$.

Flight Condition Variations--We noted earlier that $\bar{\sigma}[L]$ due to operating point changes becomes quite large at low frequencies because the helicopter's slow modes are not stable at all operating points. At intermediate and high frequency ranges, however, the uncertainty bounds are reasonably small (Fig. 3). This suggests that if the loop transfer matrix $G_0(s)$ has sufficient low frequency gain to stabilize the slow modes under all conditions, then the design might well be stable even though the (sufficient) stability-robustness condition fails. This is in fact the case. Both trial design No. 2 and No. 3 remain stable at 8 representative flight conditions ranging from hover to 160 knot forward speed and from +2000 ft/min to - 2000 ft/min ascent rates. The intuitive idea which underlies this result (sufficiently high low frequency gain) may well provide needed insight toward a generalized multivariable robustness theory for unstable perturbations.

VI. CONCLUSIONS

This paper has presented several trial control law designs to explore and illustrate the role which singular value analyses might play in multivariable system design. The examples confirm that stability-robustness condition (2) provides a reliable measure of robustness, and they show that this condition offers a natural way to limit multivariable bandwidth during the design process. The bounding formulas (5)-(8) were seen to be very useful for bounding the functions $\bar{\sigma}[L]$, although for some situations (e.g., flight condition variation) they tend to be excessively conservative.

Major weaknesses displayed by the present singular value stability-robustness theory include its inability to handle unstable perturbations and its implicit tendency to limit all loop bandwidths to be consistent with the worst case L direction. Although such situations did not arise with the helicopter, it is easy to imagine design problems where some control directions have large L's at low frequencies, hence calling for low bandwidth, while other direction have smaller L's, hence permitting greater bandwidths. It is then unduly conservative to restrict all direction to the slowest case. It is hoped that the examples presented here will help to motivate further research to overcome these weaknesses.

---

*ε is at the designers discretion

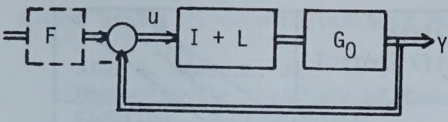


Fig. 1 Multivariable Feedback Loop

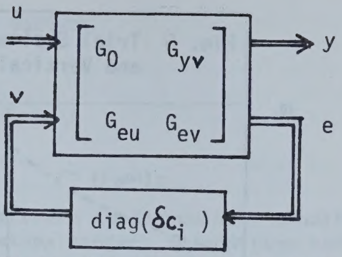
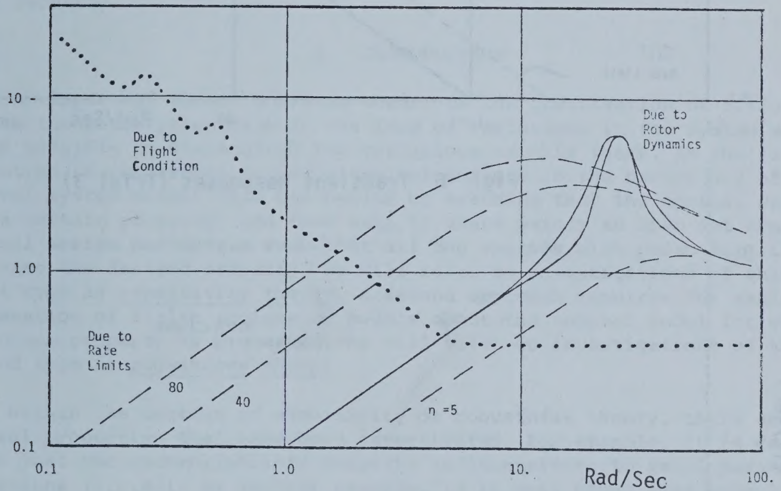

u
G_0
G_{yv}
v
G_{eu}
G_{ev}
y
e
diag(δc_i)

Fig. 2 Plant with Component Perturbation

Fig. 3 σ(L) Bound



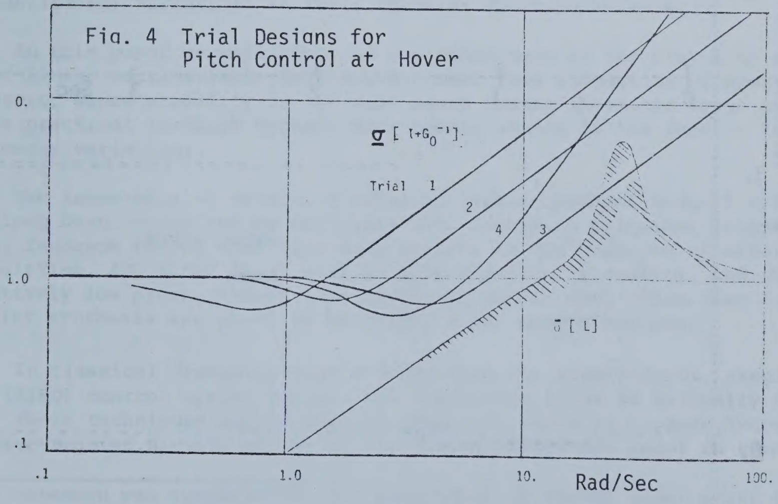


Fig. 4 Trial Designs for Pitch Control at Hover

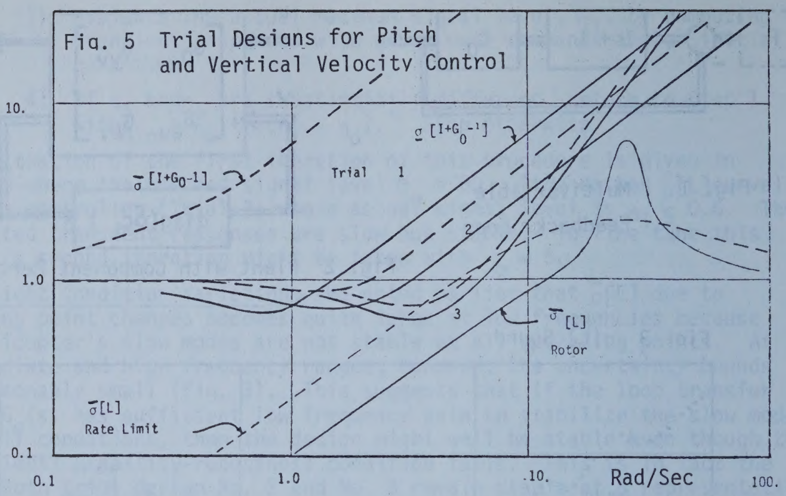
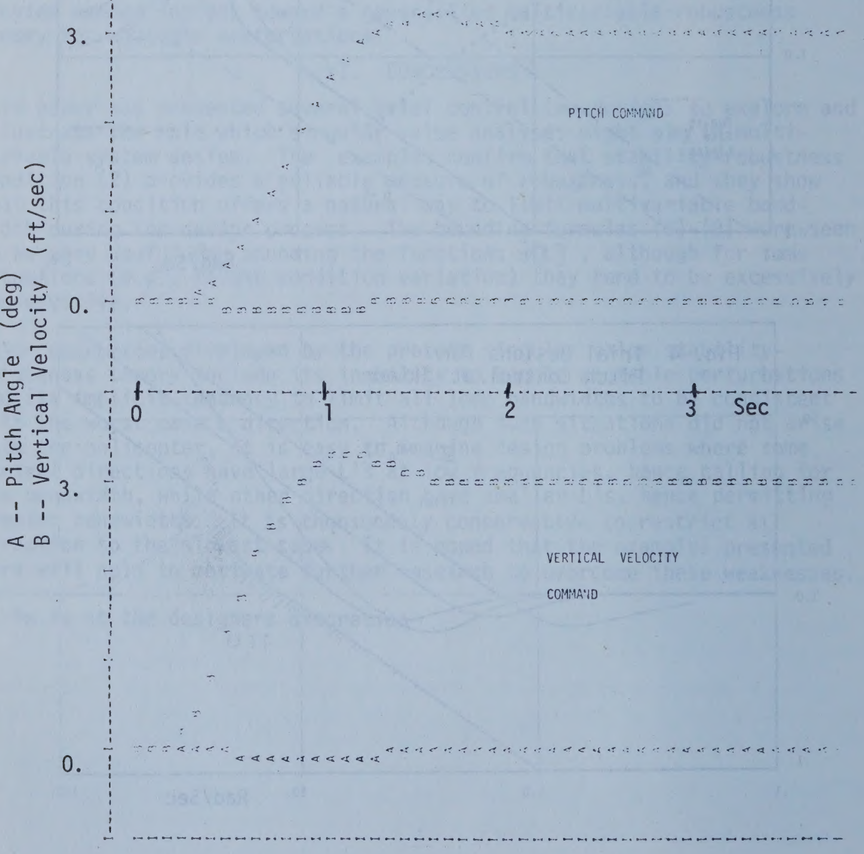


Fig. 5 Trial Designs for Pitch and Vertical Velocity Control

Fig. 6 Transient Responses (Trial 3)



ROBUST STABILITY OF MULTIVARIABLE FEEDBACK SYSTEMS

NILS R. SANDELL, JR.
Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

In this paper we will give a simple approach to determining conditions for stability of multivariable feedback systems subject to additive and multiplicative perturbations in the operators describing these systems. The approach is based on standard techniques used in input-output stability theory, and provides an alternative development and generalization of some conditions for the time-invariant case that have appeared in the literature very recently.

I. INTRODUCTION

An important theme in system theory is the preservation of various system theoretic properties in the face of variations in the system model. It is possible to distinguish two variations on this theme. In the first, attention is restricted to infinitesimal changes in the parameters of the nominal system model. Thus one begins by assuming that the nominal system has a certain property, and then asks if there exists an open set about the nominal system parameters such that all the systems with parameters in this set have the desired property. We will refer to investigations of this first type as sensitivity theory. A second approach requires the explicit delineation of finite regions of models about the nominal model for which the given property is preserved. We will refer to investigations of this second type as robustness theory.

Within the context of sensitivity or robustness theory, there are several properties that have been investigated. For example, it is well known that the controllability property is insensitive to small parameter variations [1,p.43]. As another example, it is well known that type-1 servomechanisms have zero steady state step tracking error despite large (but not destabilizing) variations in their transfer function matrices [2].

In this paper we will focus on the robustness of the stability property of linear multivariable feedback systems. This subject is of special interest, since stability is the most basic system theoretic issue and since practical feedback systems must remain stable in the face of large parameter variations.

The importance of obtaining robustly stable feedback control systems has long been recognized by designers [3]. Indeed, a principal reason for using feedback rather than open-loop control is the presence of model uncertainties. Any model is at best an approximation of reality, and the relatively low order, linear time-invariant models most often used for controller synthesis are bound to be rather crude approximations.

In classical frequency domain techniques for single-input, single-output (SISO) control system design, the robustness issue is naturally handled [3]. These techniques employ various graphical means (e.g., Bode, Nyquist, inverse-Nyquist, Nichols plots) of displaying the system model in terms of

---

This research was supported by the Department of Energy under grant ERDA-E (49-18)-2087 and by ONR under contract N00014-76-C-0346

its frequency response. From these plots, one can determine by inspection the minimum change in the model frequency response that leads to instability. These changes are often quantified by the gain and phase margins of the feedback system; sometimes the design is required to have certain minimal margins in order to be acceptable [4,p.43].

In modern time domain techniques (such as the pole placement or linear-quadratic-Gaussian approaches) for multiple-input, multiple-output (MIMO) systems, the robustness issue is not directly dealt with. Instead, it is necessary to transform the resulting design to the frequency domain to examine its robustness properties. For SISO systems, this is accomplished as for classical designs, but the situation is less clear in the MIMO case, where it is necessary to consider simultaneous variations in the frequency responses of all the loops.

Very recently, there has been some important work addressing the multivariable robustness issue. In his thesis [5,6] Safonov gives a powerful approach, based on a multivariable sector stability theorem, that can characterize robustness for very general nonlinear MIMO feedback systems. In a recent paper [7], Doyle develops a robustness characterization for the linear time invariant MIMO case. (It can be shown that Doyle's result can also be obtained by Safonov's approach [8].) Doyle's characterization involves computing the minimum singular value of a certain transfer function matrix, and this computation essentially determines the minimum simultaneous variation of the system frequency responses that leads to instability. Since there is sophisticated and widely accessible software to compute singular values [9], this characterization is of great practical value.

The present paper is prompted by the following observation. The use of singular values to characterize robustness is suggestive of connections with numerical analysis, where robust solution of equations is of interest. It would seem by adopting the framework of input-output stability, where stability is considered as a property of solutions of the feedback equation, that it should be possible to give an alternative development and generalization of Doyle's result in the spirit of numerical analysis. We shall see in the sequel that this program can in fact be readily carried out.

Notation

We will use the standard notation of input-output stability theory: see [1], pp.12-23.

$\mathcal{X}$ = some Banach space of functions $x: T \to X$ with norm $||\cdot||$

T = subset of the real numbers

X = finite dimensional vector space

$\mathcal{X}_e = \{x : P_\tau x \in \mathcal{X} \text{ for all } \tau \in T\}$

$$(P_\tau x)(t) = \begin{cases} x(t) & t \leq \tau \\ 0 & t > \tau \end{cases}$$

$\mathscr{L}_2^m$ = space of m-vector functions on T with integrable Euclidean norm

$I : \mathcal{X}_e \to \mathcal{X}_e$ = identity operator

$G : \mathcal{X}_e \to \mathcal{X}_e$ is causal if $P_\tau G P_\tau = P_\tau G$ for all $\tau \in T$

A^* = conjugate transpose of a complex matrix A

$$||G||_\Delta = \sup_{\substack{x_1,\, x_2 \,\varepsilon\, X_e \\ \tau \varepsilon T \\ P_\tau x_1 \neq P_\tau x_2}} \frac{||P_\tau G x_1 - P_\tau G x_2||}{||P_\tau x_1 - P_\tau x_2||}$$

$$g\colon X \to X, \qquad g_x = [(g_x)_{ij}] = \left[\frac{\partial g_i}{\partial x_j}\right]$$

II. ROBUST STABILITY OF LINEAR SYSTEMS

We consider the feedback system depicted in Figure 1. Here the causal operator $G\colon \mathscr{L}^m_{2e} \to \mathscr{L}^m_{2e}$ represents the plant plus any compensation that is used. The basic feedback equation is

$$(I + G)e = u \tag{1}$$

and the basic stability question is whether $(I + G)^{-1} : \mathscr{L}^m_{2e} \to \mathscr{L}^m_{2e}$ exists, is causal, and is a bounded operator in the sense that $||(I+G)^{-1}||_\Delta < \infty$. We will assume that the nominal system is stable throughout this section. We are interested in whether the closed loop system retains these properties when subject to additive (Figure 2) or multiplicative (Figure 3) perturbations representing uncertainty in the dynamic behaviour of the system.

The following theorem provides the basis for our analysis.

THEOREM 1

Let $A : X_e \to X_e$ be a causal operator, and suppose A^{-1} exists, is causal, and $||A^{-1}||_\Delta < \infty$. Then, if $\Delta A : X_e \to X_e$ is a causal operator satisfying $||\Delta A||_\Delta < \infty$ and if

$$||A^{-1}\Delta A||_\Delta < 1, \tag{2}$$

it follows that $(A + \Delta A)^{-1} : X_e \to X_e$ exists, is causal and has

$$||(A+\Delta A)^{-1}||_\Delta < \infty \tag{3}$$

Proof

By Theorem 2.10 of [12], $(I + A^{-1}\Delta A)^{-1}$ exists, is causal and satisfies $||I + A^{-1}\Delta A||_\Delta < \infty$. Since

$$(A + \Delta A)^{-1} = (I + A^{-1}\Delta A)A^{-1} \tag{4}$$

the theorem follows. Q.E.D.

Remarks

1. Since

$$||A^{-1}\Delta A||_\Delta \leq ||A^{-1}||_\Delta \; ||\Delta A||_\Delta \tag{5}$$

a sufficient condition for (2.1) is

$$||A^{-1}||_\Delta \; ||\Delta A||_\Delta \; < \; 1. \qquad (6)$$

2. A basic result in numerical analysis is that if an nxn matrix A is invertible, then A + ΔA is invertible for all ΔA satisfying

$$\overline{\sigma}(\Delta A) \; < \; \underline{\sigma}(A) \qquad (7)$$

where $\overline{\sigma}(\Delta A) = ||\Delta A||_2$, $\underline{\sigma}(A) = (||A^{-1}||_2)^{-1}$ are respectively the largest and smallest singular values of the matrix A. Theorem 1 is thus a generalization of this classical finite dimensional result where, of course, boundedness and causality are not at issue. In the finite dimensional case there exists ΔA such that $\overline{\sigma}(\Delta A) = \underline{\sigma}(A)$ and A + ΔA is singular;this is easily proved by the singular value decomposition.[1]

3. Theorem 1 can be used to give $\mathscr{L}_p$ robust stability results, but we will confine ouselves to the case p = 2 in the sequel.

4. Theorem 1 can be used to obtain robust stability results for both continuous and discrete time, but in the sequel we will confine our attention to the case T = [0,∞].

The robust stability questions posed at the beginning of this section are now answered in terms of Theorem 2.

THEOREM 2

Assume that the basic feedback system of Figure 1 is stable. Then (i) the system remains stable for additive perturbations ΔG(Figure 2)provided

$$||(I+G)^{-1}\Delta G||_\Delta \quad < \quad 1 \qquad (8)$$

and (ii) the system remains stable for multiplicative perturbations ΔG (Figure 3) provided

$$||[I-(I+G)^{-1}]\Delta G||_\Delta \quad < \quad 1 \qquad (9)$$

Proof

For case (i), we apply Theorem 1 to the equation

$$(I+G+\Delta G)e \; = \; u \qquad (10)$$

while for case (ii) we consider

$$[I+G(I+\Delta G)]e \; = \; (I+G+(I+G)\Delta G-\Delta G)e \; = u. \quad \text{Q.E.D.} \quad (11)$$

The practical importance of Theorem 2 stems from the fact that the norm of a linear convolution operator or a memoryless nonlinear operator can be readily computed.

---

[1] The singular value decomposition of an nxn nonsingular complex matrix A is $A=U\Sigma V^*$, where U and V are unitary nxn matrices, $\Sigma=\text{diag}(\sigma_1,\ldots,\sigma_n)$ and the singular values σ_i are the non-negative square roots of the eigenvalues of A*A. See [14] for references, a more general definition,and an excellent discussion of the fundamental role of the singular value decomposition in linear systems theory.

Lemma 1 [13, p. 26]

Let the operator $G: \mathscr{L}^m_2 \to \mathscr{L}^m_2$ for $T = [0,\infty]$ be defined by

$$(Gx)(t) = \int_0^\infty G(t-\tau)\ x(\tau)d\tau \tag{12}$$

where the elements of the impulse response matrix $G(t)$ are assumed absolutely integrable on T. Then

$$||G||_\Delta = ||G||_{\mathscr{L}^m_{2e}} = \sigma_{max} \tag{13}$$

where

$$\sigma_{max} = \max_{\omega \geq 0}\ \max_{1 \leq i \leq m}\ \sigma_i(G(j\omega)) \tag{14}$$

and where $\sigma_i(G(j\omega))$ denotes the ith singular value of the transfer function matrix corresponding to G.

Lemma 2 [12, p. 166]

Let the operator $G: \mathscr{L}^m_{2e} \to \mathscr{L}^m_{2e}$ for $T = [0,\infty]$ be defined by

$$(Gx)(t) = g(x(t)) \tag{15}$$

where $g: R^m \to R^m$ is continuously differentiable. Then

$$||G||_\Delta = \sup_{x \in R^m} \bar{\sigma}(g_x(x)). \tag{16}$$

Combining Theorem 2 and Lemmas 1 and 2, we obtain the following result.

THEOREM 3

Assume that the nominal system in Figure 1 is time invariant, stable, and that the operator $(I+G)^{-1}$ can be represented as a convolution operator whose impulse response matrix has absolutely integrable elements. Assume that (Case a) ΔG is a stable convolution operator whose impulse response matrix has absolutely integrable elements or (Case b) ΔG is a memoryless nonlinear operator induced by the continuously differentiable function $\Delta g: R^m \to R^m$. Then (i) the system remains stable for additive perturbations satisfying

$$\begin{aligned} &\text{Case a:} \quad \bar{\sigma}(\Delta G(j\omega)) < \underline{\sigma}(I+G(j\omega)), \quad \omega > 0 \\ &\text{Case b:} \quad \bar{\sigma}(\Delta g_x(x)) < \underline{\sigma}(I+G(j\omega)), \quad \omega > 0, \quad x \in R^m, \end{aligned} \tag{17}$$

and (ii) the system remains stable for multiplicative perturbations satisfying

$$\begin{aligned} &\text{Case a:} \quad \bar{\sigma}(\Delta G(j\omega)) < \underline{\sigma}(I+G^{-1}(j\omega)), \quad \omega > 0 \\ &\text{Case b:} \quad \bar{\sigma}(\Delta g_x(x)) < \underline{\sigma}(I+G^{-1}(j\omega)), \quad \omega > 0, \ x \in R^m. \end{aligned} \tag{18}$$

Here $G(j\omega)$ and $\Delta G(j\omega)$ are the transfer functions of G and ΔG, and $\bar{\sigma}(A)$ and $\underline{\sigma}(A)$ denote the maximum and minimum singular values of A.

PROOF

(i) From Lemmas 1 and 2 and (17) we have in either case

$$||\Delta G|| < ||(I+G)^{-1}||^{-1} \qquad (19)$$

so that

$$||(I+G)^{-1}||\ ||\Delta G|| < 1. \qquad (20)$$

(ii) Note that

$$\underline{\sigma}(I+G^{-1}(j\omega)) = \{\overline{\sigma}[(I+G^{-1}(j\omega))^{-1}]\}^{-1}$$

$$= \{\overline{\sigma}[(I+G(j\omega))^{-1}G(j\omega)]\}^{-1} = \{\overline{\sigma}[I-(I+G(j\omega))^{-1}]\}^{-1} \qquad (21)$$

Therefore Lemmas 1 and 2 and (18) imply that

$$||I-(I+G)^{-1}||\ ||\Delta G|| < 1. \qquad (22)$$

Q.E.D.

Remarks

1. Notice the analogy between Theorem 3 concerning robust stability of linear systems and the classical result quoted previously concerning robust inversion of matrices (or bounded operators).

2. Theorem 3 for the case of rational transfer function matrices is the result of Doyle alluded to previously. Doyle's proof is completely different, however, depending on the multivariable Nyquist Theorem, so that the connections with the inversion issue are only implicit.

3. The quantity $\underline{\sigma}(I+G(j\omega))$ or $\underline{\sigma}(I+G^{-1}(j\omega))$ is easily computed and plotted as a function of ω. Doyle has made great use of this technique in the analysis of multivariable feedback systems. Such a plot plays much the same role for determining MIMO robustness properties as the more classical Bode, etc. plots in SISO design.

4. The quantity $\underline{\sigma}^{-1}(I+G(j\omega))$ is the generalization of the classical Bode SISO sensitivity function of changes in the closed-loop transfer function with respect to changes in the open-loop transfer function in the following sense. Let y_1 denote the output of the system of Figure 1 for a given input and y_2 the corresponding output of the system of Figure 2 for the same input. Then we can show

$$y_1(j\omega) - y_2(j\omega) = (I+G(j\omega))^{-1}\Delta G(j\omega)\ (G(j\omega)+\Delta G(j\omega))^{-1}y_2(j\omega) \qquad (23)$$

so that

$$||y_1(j\omega) - y_2(j\omega)|| \leq \frac{1}{\underline{\sigma}(I+G(j\omega))}||\Delta G(j\omega)\ (G(j\omega)+\Delta G(j\omega))^{-1}|| \times$$

$$||y_2(j\omega)||. \qquad (24)$$

Consequently, the percentage change in the closed-loop transfer function matrix is attenuated from the percentage change in the open loop transfer function matrix by the factor $\underline{\sigma}^{-1}(I+G(j\omega))$.

5. On the other hand, the perturbations defining the SISO gain and phase margins are multiplicative rather than additive, so that $\underline{\sigma}(I+G^{-1}(j\omega))$ is more appropriate as a measure of the tolerance of the feedback system to model uncertainty.

6. It is in general impossible to express $\underline{\sigma}(I+G^{-1}(j\omega))$ in terms of $\underline{\sigma}(I+G(j\omega))$.

7. Consider the feedback system in Figure 1 with

$$G(s) = -G(sI - A)^{-1}B \qquad (25)$$

where

$$G = B'K \qquad (26)$$

$$O = A'K + KA + C'C - KBB'K \qquad (27)$$

(We assume [A,B] controllable and [A,C] observable so that a unique positive definite solution of (27) exists.) The well known equality [15]

$$(I+G(-sI-A)^{-1}B]' \; [I+G(sI-A)^{-1}B] = I+[C(-sI-A)^{-1}B]'x$$

$$[C(sI-A)^{-1}B] \qquad (28)$$

which follows from (27) after a little manipulation, shows that the system is robust to additive perturbations. Moreover, it can be inferred from a more general result of Safonov and Athans [6] that the system of Figure 3 with G(s) defined by (25) is stable for $\Delta G(j\omega)$ satisfying

$$\overline{\sigma}(\Delta G(j\omega)) \leq 1/2 \text{ for all } \omega \geq 0;$$

this can also be inferred directly using the inequality [17]

$$\underline{\sigma}(I+G^{-1}(j\omega)) \geq \frac{\underline{\sigma}(I+G(j\omega))}{1+\underline{\sigma}(I+G(j\omega))} \qquad (29)$$

together with (28) and Theorem 3.

III. SUMMARY AND CONCLUSIONS

In this paper we have considered the robustness of the stability property of a linear feedback system to variations in the system model. The approach was to generalize a fundamental result in numerical linear algebra concerning robust inversion of matrices to operators of the type arising in input-output stability theory. In the time-invariant case, this approach specializes to give sufficient conditions for stability under additive and multiplicative perturbations that are easily verified by computing the singular values of certain transfer function or partial derivative matrices.

The results of this paper are felt to be of interest for two reasons. First, as Doyle has previously pointed out in the time invariant case [7], the characterization and design of robustly stable MIMO feedback systems is a fundamental problem in control theory that has yet to be completely resolved. Second, as has been previously emphasized by Zames [16] and Safonov [5], a fundamental problem in large scale system theory is to give conditions for the success of designs based on multiple, aggregate models of a single large system - this is essentially a robustness problem.

Acknowledgement

The ideas in this paper arose during the course of discussions with M. Athans, J.C. Doyle, A.J. Laub, M. Safonov, and G. Stein.

Part of the work for this paper was accomplished during a visit to the Centro di Studio dei Sistemi di Controllo e Calcole Automatica of the University of Rome; the author gratefully acknowledges the hospitality of Prof. F. Nicolo.

References

1. W.M. Wonham: Linear Multivariable Control: A Geometric Approach. Springer-Verlag, New York (1974).

2. N.R. Sandell, Jr. and M. Athans: On Multivariable type-ℓ systems, Automatica 9, pp. 131-136 (1973).

3. I.M. Horowitz: Synthesis of Feedback Systems. Academic Press, New York (1963).

4. BACKGROUND INFORMATION AND USER GUIDE FOR MIL-F-9490D, Flight Control Systems - Design, Installation and Test of Piloted Aircraft, General Specification for, Air Force Flight Dynamics Laboratory, Technical Report AFFDL-TR-74-116, January 1975.

5. M. G. Safonov: Robustness and Stability Aspects of Stochastic Multivariable Feedback System Design. Ph.D. Thesis, Massachussetts Institute of Technology (1977).

6. M. Safonov and M. Athans: Gain and Phase Margins for Multiloop LQG Regulators. IEEE Trans. on Aut. Control AC-22, pp. 173-179 (1977).

7. J. C. Doyle: Robustness of Multiloop Linear Feedback Systems. To appear.

8. M. G. Safonov, personal communication.

9. B. S. Garbow et. al.: Matrix Eigensystem Routines - EISPACK Guide Extension. Lecture Notes in Computer Science, Vol. 51, Springer-Verlag, New York (1977).

10. N.R. Sandell, Jr., P. Variaya, M. Athans, and M. G. Safonov: Survey of Decentralized Control Methods for Large Scale Systems. IEEE Trans. on Aut. Control, AC-23, pp. 108-128 (1978).

11. H. K. Khalil and P. V. Kokotovic: Control Strategies for Decision Makers Using Different Models of the Same System. IEEE Trans. on Aut. Control, AC-23, pp. 289-298 (1978).

12. J.C. Willems, The Analysis of Feedback Systems. M.I.T. Press, Cambridge, Mass. (1971).

13. C. A. Desoer and M. Vidyasagar: Feedback Systems: Input-Output Properties. Academic Press, N.Y. (1975).

14. A. J. Laub: Linear Multivariable Control. Numerical Considerations. Invited paper, American Mathematical Society Short Course on Control Theory, Providence, R.I. (1978). Also ESL-P-833 Electronic Systems Laboratory, Massachusetts Institute of Technology.

15. R.E. Kalman: When is a Linear System Optimal? Trans. ASME Ser.D: J. Basic Eng., Vol. 86, pp. 51-60 (1964).

16. G. Zames: On Feedback Hierarchies and Complexity (or information), 1976 CDC, Clearwater, Florida (1976).

17. D. W. Nuzman and N.R. Sandell, Jr., "An Inequality Arising in Robustness Analysis of Multivariable Systems", submitted for publication.

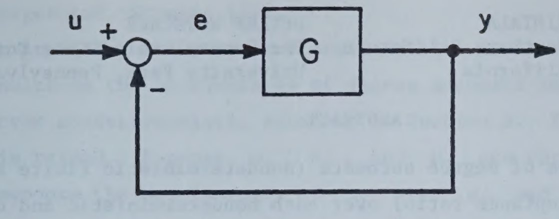


Figure 1. Basic MIMO Linear Feedback System

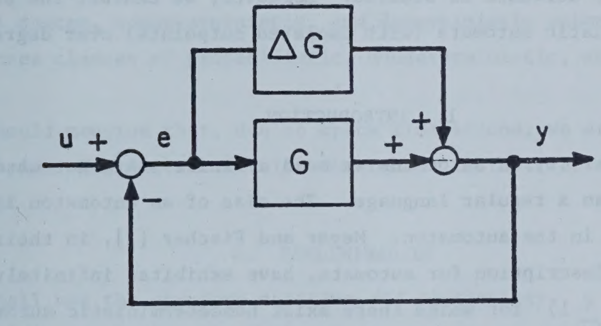


Figure 2. System Subject to Additive Perturbations

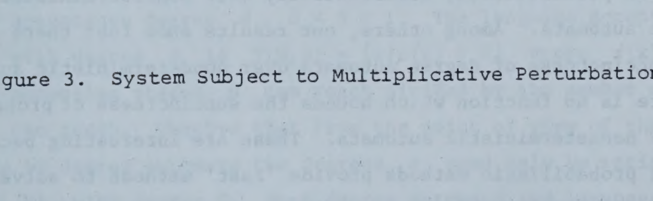


Figure 3. System Subject to Multiplicative Perturbations

SUCCINCTNESS OF DEGREE AUTOMATA AND PROBABILISTIC AUTOMATA[*] Preliminary Version

CHANDRA M. R. KINTALA
University of Southern California
Los Angeles, California

DETLEF WOTSCHKE
The Pennsylvania State University
University Park, Pennsylvania

ABSTRACT

The succinctness of degree automata (nondeterministic finite automata with a specified acceptance ratio) over both nondeterministic and deterministic finite automata is investigated. Similarly, the succinctness of probabilistic automata with isolated cutpoints over both nondeterministic and deterministic automata is studied. Moreover, we exhibit the succinctness of probabilistic automata (with isolated cutpoints) over degree automata.

1. INTRODUCTION

In this paper, by an automaton we mean a finite-state automaton and by a language we mean a regular language. The size of an automaton is the number of states in the automaton. Meyer and Fischer [1], in their paper on economies of description for automata, have exhibited infinitely many languages $\{A_n | n \geq 1\}$ for which there exist nondeterministic automata of size n, but the equivalent minimal deterministic automata are of size 2^n. Thus, nondeterministic automata can provide descriptions of languages which are exponentially-succinct (2^n-succinct) over the descriptions by deterministic automata. Schmidt [3] recently proved the $2^{\sqrt{n}}$-succinctness of nondeterministic automata over unambiguous automata and also that of unambiguous automata over deterministic automata.

We study here the succinctness of degree automata (nondeterministic automata with a specified acceptance degree d, $0 \leq d \leq 1$) and that of probabilistic automata (nondeterministic automata whose transitions are associated with probabilities) simultaneously over nondeterministic and deterministic automata. Among others, our results show that there is a polynomial-succinctness of degree automata over nondeterministic automata and that there is no function which bounds the succinctness of probabilistic automata over nondeterministic automata. These are interesting because it is known that probabilistic methods provide 'fast' methods to solve some 'seemingly intractable' problems with a large degree of confidence in their answers [6]. Similarly our results suggest that probabilistic methods might be very efficient in lexical analyzers since regular expressions are

---

[*] This research was supported in part by the National Science Foundation under Grant No. MCS76-10076A01.

heavily used in connection with the specification of the syntax of programming languages (cf. Schmidt [3]).

After providing the necessary preliminaries in Section 2, we first obtain results on the succinctness of degree automata and probabilistic automata over nondeterministic automata in Section 3. Then the following question is raised. Suppose μ_1, μ_2, and μ_3 are three classes of automata and suppose the succinctness of μ_1 over μ_2 and that of μ_2 over μ_3 are known. Is there then a sequence of languages for which μ_1 is succinct over μ_2 and μ_2 is succinct over μ_3 for the same sequence? We answer this question, in Section 4, in the affirmative for the three classes of degree, nondeterministic, and deterministic automata and also for the three classes of probabilistic, nondeterministic, and deterministic automata.

We should mention that, due to space limitations, we are omitting some of the proofs. For further details the reader is referred to the technical report [5].

2. PRELIMINARIES

We shall use the standard notation and terminology. ε is the empty word and $|x|$ is the length of the string x. We assume the reader is familiar with the standard definitions of a nondeterministic finite automaton, NFA, $M = (Q,\Sigma,\delta,q_0,F)$, a configuration (q,x) of M for $q \varepsilon Q$ and $x \varepsilon \Sigma^*$, the transition relations $\vdash$ and $\overset{*}{\vdash}$ between configurations, and the language $T(M)$ accepted by M. M is a deterministic finite automaton, DFA, if for all $q \varepsilon Q$ and $a \varepsilon \Sigma$, $\delta(q,a)$ contains at most one element. Thus we do not require the deterministic automata to be completely specified. M is said to be an unambiguous finite automaton, UFA, if no word is accepted in more than one way.

A degree automaton, DA, is a nondeterministic automaton along with a specified acceptance degree d, $0 \leq d \leq 1$. The language accepted by such a DA M with degree d is $T(M,d) = \{x | r(x) > d\}$ where $r(x)$ is the number of accepting states x can reach divided by the number of all states which x can reach. Observe that from the point of view of the languages acceptable by degree automata the degrees d need only be rational. Every NFA is a DA with degree 0. Such degree automata and languages have been studied with respect to general classes of automata by Wotschke [4]. It is obvious that the canonical algorithm for converting any NFA into an equivalent DFA can essentially be used to convert any DA into an equivalent DFA. Thus degree automata accept only regular sets. However, our study will show that degree automata can be polynomially succinct over NFA's which in turn are exponentially succinct over DFA's for the same languages infinitely often.

A probabilistic automaton, PA, is a nondeterministic automaton whose transitions are associated with probabilities and which has a specified cutpoint λ, $0 \leq \lambda < 1$. Informally, a word w will be accepted by such a PA M with cutpoint λ iff the sum of the probabilities of all paths, reaching the final states of M on word w and starting at the specified initial state, is $> \lambda$. Formally, a probabilistic automaton is a pair (M,λ) where λ is a cutpoint, $0 \leq \lambda < 1$, and M is a 5-tuple $(Q,\Sigma,\{A(a) \mid a\varepsilon\Sigma\}, q_0,F)$ where

$Q = \{q_0,q_1,\ldots,q_{k-1}\}$ is a set of k states,

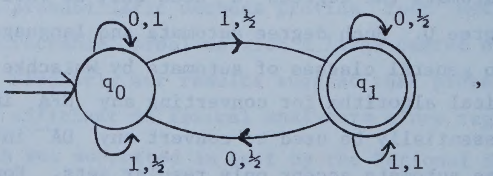
Σ is a finite alphabet,

$F \subseteq Q$ is the set of final states

$A(a)$ is a stochastic matrix of order k associated with $a\varepsilon\Sigma$. (A stochastic matrix A of order k is a matrix $[a_{ij}]$ of size $k \times k$ such that $0 \leq a_{ij} \leq 1$ and $\sum_j a_{ij} = 1$ for all i.)

If $x\varepsilon\Sigma^*$, such that $x = a_1a_2\ldots a_n$, then $A(x) = A(a_1)A(a_2)\ldots A(a_n) = [a_{ij}(x)]$. Thus, $a_{ij}(x)$ is the transition probability from state q_i to state q_j associated with the word x. If $F = \{q_{i_1},\ldots,q_{i_s}\}$ then let U_F denote the column vector which contains 1's at the $i_1,\ldots,i_s$ positions and 0's elsewhere. Let U_0 denote the unit-row vector $[1\ 0\ 0\ \ldots\ 0]$ of size k. Then define, for all $x\varepsilon\Sigma^*$, $P_M(x) = U_0A(x)U_F$. Then $P_M(x)$ is the probability of M reaching one of the final states on the string x starting at state q_0. The language accepted by such a PA (M,λ) is $T(M,\lambda) = \{w \mid P_M(w) > \lambda\}$. For the sake of clarity, we shall describe PA's by their state diagrams whose arcs are labelled with the input symbol and the corresponding probability. We illustrate the above concepts by the following example from Paz [2].

Example: (M,λ) is a PA where $M = (\{q_0,q_1\}, \{0,1\}, \{A(0),A(1)\}, q_0, \{q_1\})$ where $A(0) = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ and $A(1) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$. Such a PA is described by the following diagram.



, cutpoint λ

It is easy to see that for $x = a_1 \ldots a_k$, $P_M(x) = .a_k \ldots a_1$ in binary representation. So if $0 < \lambda_1 < \lambda_2 < 1$, then $T(M,\lambda_1) \neq T(M,\lambda_2)$. Since there are nondenumerably many such λ, this example shows the existence of a PA accepting nonregular sets. Paz uses this example to prove the existence of regular languages for which there exist 2-state PA's but for which the sizes of the equivalent DFA (or NFA) are not bounded. Thus PA's are nonrecursively-succinct over NFA's and DFA's. In light of the fact that PA's accept non-regular languages this result might be not too surprising. However, Rabin has shown that PA's with isolated cutpoints accept all and only the regular sets. So we will investigate whether probabilistic automata with isolated cutpoints can also provide arbitrary savings in the numbers of states over nondeterministic automata.

To be more specific, let (M,λ) be a PA. The cutpoint λ is ε-isolated with respect to M if $|P_M(x) - \lambda| \geq \varepsilon$ for all $x \varepsilon \Sigma^*$ for some $\varepsilon > 0$. Then Rabin (cf. Paz [2]) proved the following.

Theorem 2.1: If λ is an ε-isolated cutpoint for a PA (M,λ), then there exists a DFA N such that $T(M,\lambda) = T(N)$. If M has n states, then N can be chosen to have m states where $m \leq (1 + \frac{1}{2\varepsilon})^{n-1}$.

We exhibit here infinitely many languages for which there exist 2-state PA's with isolated cutpoints, but the equivalent NFA's have no bound on their sizes. Observe that the examples used by Paz do not have isolated cutpoints.

Definition 2.2: A class of automata μ_1 is said to be f(n)-succinct over another class μ_2 for a sequence of languages L_n, $n \geq 1$, if there is an n-state automaton $M_1 \varepsilon \mu_1$ accepting L_n, and every $M \varepsilon \mu_2$ accepting L_n must have at least $f(n)$ states for almost all n. By $\mu_1 \xrightarrow{f(n)} \mu_2$ we denote the f(n)-succinctness of μ_1 over μ_2 for some sequence of languages.

Let PA, DA, NFA, UFA, and DFA also denote the classes[1)] of probabilistic, degree, nondeterministic, unambiguous, and deterministic finite automata, respectively. It is already known that

(i) $\text{NFA} \xrightarrow{2^n} \text{DFA}$

and

(ii) $\text{NFA} \xrightarrow{2^{c\cdot\sqrt{n}}} \text{UFA}$ and $\text{UFA} \xrightarrow{2^{c\cdot\sqrt{n}}} \text{DFA}$

(Schmidt [3]). We shall obtain results on the succinctness of PA and DA over NFA and DFA. Given three classes of automata μ_1, μ_2, and μ_3, we say $\mu_1 \xrightarrow{f(n)} \mu_2 \xrightarrow{g(n)} \mu_3$ if there is a sequence of languages

---

[*] The context will always make clear whether DA, e.g., denotes a single degree-automaton or the class of all degree-automata.

L_n for which $\mu_1 \xrightarrow{f(n)} \mu_2$ and $\mu_2 \xrightarrow{g(n)} \mu_3$ for the same sequence of languages. Such succinctness, we call simultaneous succinctness. We can similarly define simultaneous succinctness of four classes of automata.

3. SUCCINCTNESS OF DA AND PA

For each $n > 1$, define the language B_n over the alphabet $\{a,b,c,r\}$ as

$$B_n = \{w \mid w = w_0 r w_1 r w_2 \ldots w_{k-1} r w_k \text{ for some } k \geq 0 \text{; for each } 0 \leq i \leq k \text{, the following holds:}$$

$$\text{(i) } w_i \varepsilon \{a,b,c\}^*$$

$$\text{(ii) if } w_i = xyz \text{ for some } y \varepsilon \{c\}^* \text{ then } |y| < n$$

$$\text{(iii) if } w_i = xyaz \text{ for some } y \varepsilon \{b,c\}^* \text{ then } |y| < n$$

$$\text{(iv) if } w_i = xybz \text{ for some } y \varepsilon \{a,c\}^* \text{ then } |y| < n\}.$$

Intuitively, in each w_i of w as above, every substring of length n or more must contain an occurrence of 'a' and an occurrence of 'b'. The symbol 'r' acts like a 'reset character'. It is relatively simple to see that B_n can be accepted by an $(n^2 + 1)$ state DFA.

Since a and b are two distinct symbols, we expect that any finite automaton, deterministic or not, must internallly keep two separate counters for the non-a and non-b characters, each counter being capable of counting at least n different values. The following lemma is the formal counterpart to that intuition.

Lemma 3.1: Any nondeterministic automaton accepting B_n must have at least n^2 states.

Proof: Suppose N is an nfa accepting B_n. We shall construct a fixed string w and exhibit n^2 different positions in w in such a way that any nfa correctly accepting B_n must be in distinct states at those positions during an accepting computation on w.

For each possible pair (i,j), $0 \leq i, j \leq n-1$, define w_{ij} and v_{ij} as below:

Case(i) $i > j$: $w_{ij} = c^{i-j-1}bc^j$

$$v_{ij} = c^{n-i-1}ac^{i-j-1}r$$

(ii) $j > i$: $w_{ij} = c^{j-i-1}ac^i$

$$v_{ij} = c^{n-j-1}bc^{j-i-1}r$$

(iii) $i = j$: $w_{ij} = c^{i}$

$$v_{ij} = c^{n-i-1}r$$

Let $w = \prod_{i=0}^{n-1} \prod_{j=0}^{k-1} w_{ij} \cdot v_{ij}$, where Π denotes the iterated concatenation of the strings defined over the specified indices. The idea is to force the machine N to go to a 'state' (i,j) after reading w_{ij} from w, for all possible i,j. The strings v_{ij} are so designed that in $w_{ij}v_{ij}$ the number of non-a characters before the next occurrence of a (or the reset character) is exactly $n - 1$, and similarly for b. Observe that $w \varepsilon B_n$ and hence there is an accepting computation of N on w. Consider one such computation. Let q_{ij} be the state of N after reading the initial substring of w up to and including w_{ij} in the accepting computation being considered.

Claim: $(g,h) \neq (i,j) \Rightarrow q_{gh} \neq q_{ij}$.

Proof: Assume otherwise, i.e., suppose there exist (g,h) and (i,j) such that $(g,h) \neq (i,j)$ but $q_{gh} = q_{ij}$. Without loss of generality, assume that $(g,h) < (i,j)$ using the order in w. Then decompose w such that $w = xw_{gh}v_{gh}yw_{ij}v_{ij}z$ for some $x,y,z \varepsilon \{a,b,c,r\}^{*}$. Since $q_{gh} = q_{ij}$, by assumption, both $w' = xw_{gh}v_{ij}z$ and $w'' = xw_{gh}(v_{gh}yw_{ij})^{2}v_{ij}z$ must be accepted by N. However, we claim that either w' or w'' does not belong to B_n. This is because

(1) in the string $w_{gh}v_{ij}$, the number of non-a (non-b) characters before an occurrence of a (b), or the reset character r, is $n - 1 + g - i$($n - 1 + h - j$, respectively).

(2) in the string $w_{ij}v_{gh}$, similarly, the number of non-a (non-b) characters before an occurrence of a (b), or the reset character r, is $n - 1 + i - g$($n - 1 + j - h$, respectively). Since $(g,h) \neq (i,j)$, either $g - i$, $i - g$, $h - j$, or $j - h$ must be greater than zero. So either $w_{gh}v_{ij}$ or $w_{ij}v_{gh}$ cannot be a substring in any string belonging to B_n and hence w' or w'' is not in B_n, but both are accepted by N, a contradiction.

Since there are n^2 possible q_{ij}'s, all of which are distinct by the above claim, N must have at least n^2 states. Q.E.D.

It is, however, relatively simple to see that there is a $(2n + 2)$-state degree automaton with degree ½ which accepts B_n.

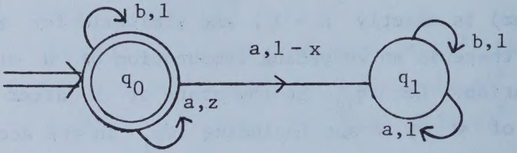
The above remark together with Lemma 3.1 gives us the following theorem on the succinctness of DA.

Theorem 3.2: DA $\xrightarrow{O(n^2)}$ NFA.

It is easy to see that the languages B_n can be generalized to show that DA $\xrightarrow{O(n^k)}$ NFA for any $k \geq 2$.

We will now investigate the succinctness of probabilistic automata (with isolated cutpoints) over nondeterministic finite automata.

Let $C_n = \{w | w \epsilon \{a,b\}^*$, w has at most n occurrences of $a\}$. The following 2-state probabilistic automaton S_n accepts C_n with an isolated cutpoint of ½.



where $x = 2^{-1/n+\beta}$ so that $x^\ell > ½$ for $\ell \leq n$ and $x^\ell < ½$ for $\ell > n$.

(Recall that the second component in the labels on the arcs denote the probabilities). It is easy to construct an $(n + 1)$-state DFA accepting C_n.

Lemma 3.3: Every NFA accepting C_n must have at least $(n + 1)$-states.

Proof: Consider the string $w = a^n$ and the sequence of states $q_0, q_1, \ldots, q_n$ entered during an accepting computation on w by an NFA N accepting C_n. All these states must be distinct for otherwise, if $q_i = q_j$ for $i \neq j$ then there will be an accepting computation by N on $a^{n+|i-j|}$ where $|i-j| > 0$, contradicting the assumption that N correctly accepts C_n. Q.E.D.

We can also show the following.

Lemma 3.4: Every degree automaton accepting C_n must have at least $\log(n + 1)$ states.

Let D be a DA accepting C_n with degree d. Let Q_i, $0 \leq i \leq n$ be the set of states reachable by D after receiving the i^{th} a in the string a^n in C_n. We claim all these sets must be distinct. For otherwise, if $Q_i = Q_j$ for $i \neq j$, then the ratio of acceptance of D on $a^{n+|i-j|}$ will be equal to that on a^n which is $> d$, a contradiction.

The above two lemmas together provide us with the following theorem on the succinctness of PA.

Theorem 3.5: There is no function which can bound the succinctness of PA over DA and also that of PA over NFA, even when the PA are provided with isolated cutpoints.

Remark: One could possibly object to our defining the size of a PA by the number of states only since the probabilities and/or cutpoints play an essential role and since their representations grow as n increases in the sequence C_n. On the other hand it is worthwhile exhibiting that almost all the information carried by states in an NFA can be encoded in the probabilities and/or cutpoints in a PA. Furthermore it should be noted that the degree-automata for C_n, which are also probabilistic automata with isolated cutpoints, have only one additional information compared to NFA's, namely the cutpoint ½ which is constant over the entire sequence C_n. Thus PA's can be at least polynomially more succinct than NFA's by any reasonable measure of succinctness.

4. SIMULTANEOUS SUCCINCTNESS OF PA AND DA

We will list only the major theorems in this section. For details the reader is referred to the technical report [5].

Theorem 4.1: For any monotonically increasing function $g(n)$ such that $g(n) \ll 2^n$, there is a sequence of languages D_n, $n > 4$, such that (i) there is a PA accepting D_n having only $O(n)$, states (ii) there is an NFA accepting D_n having $O(g(n))$ states and every NFA accepting D_n must have at least $g(n) + n$ states, and (iii) there is a DFA accepting D_n having $O(2^n)$ states and every DFA accepting D_n must have at least $(2^{m-4} + g(m))$ states.

So there are infinitely many languages for which the probabilistic automata are $g(n)$-succinct over NFA and NFA are $2^{g^{-1}(n)}$-succinct over DFA for the same sequence of languages.

We can similarly obtain the following theorem on the simultaneous succinctness of DA over NFA and DFA.

Theorem 4.2: For each $k \geq 2$ there is a sequence of languages E_n for which $\text{DA} \xrightarrow{O(n^k)} \text{NFA} \xrightarrow{O(2^{n^{1/k}})} \text{DFA}$.

Moreover we can obtain the following theorem.

Theorem 4.3: For any monotonically increasing function $g_1(n)$ such that $g_1(n) \ll (n^k) \ll 2^n$, for a given $k > 1$, there exists a sequence of languages for which $\text{PA} \xrightarrow{O(g_1(n))} \text{DA} \xrightarrow{O((g_1^{-1}(n))^k)} \text{NFA} \xrightarrow{2^{g^{-1}(n)}} \text{DFA}$.

Conclusion: Our results in this paper show that probabilistic automata with isolated cutpoints and degree automata, though they accept only regular sets, are useful and practical concepts since they often can characterize regular sets much more concisely than nondeterministic automata can.

REFERENCES

[1] Meyer, A. R. and M. J. Fischer (1971). "Economy of Description by Automata, Grammars and Formal Systems," Conference Record, IEEE 12th Annual Symposium on Switching and Automata Theory, pp. 188-190.

[2] Paz, A. (1971). "Introduction to Probabilistic Automata," Academic Press, New York.

[3] Schmidt, E. M. (1977). "Succinctness of Description of Context-free, Regular and Finite Languages," Ph.D. Thesis, Cornell University.

[4] Wotschke, D. (1977). "Degree Languages: A New Concept of Acceptance," JCSS, V. 14, N 2, pp. 187-299.

[5] Kintala, C. M. R. and D. Wotschke (1978). "Succinctness of Degree Automata and Probabilistic Automata," PSU Technical Report.

[6] Rabin, M. O. (1976). "Probabilistic Algorithms," in Algorithms and Complexity: New Directions and Recent Results, J. F. Traub, ed., Academic Press, New York, 1976, pp. 21-39.

ON LANGUAGES ACCEPTED BY TIME-BOUNDED ORACLE MACHINES[†]

RONALD V. BOOK AND CELIA WRATHALL[††]
Department of Mathematics
University of California at Santa Barbara
Santa Barbara, California 93106

INTRODUCTION

In [3] it is shown that a number of different classes of languages can be characterized in terms of the class of regular sets, the Boolean operations, and certain restrictions on the operation of homomorphic replication. These characterizations are simple and uniform and reveal that the various classes have extremely similar structures. In this paper we consider classes of languages specified by relative acceptance, that is, classes of languages accepted by Turing machines with oracles; in particular, we characterize various classes of languages specified by nondeterministic time-bounded oracle machines. Again we rely on the class of regular sets as a sub-basis and we use the Boolean operations and certain restrictions on homomorphic replication; since we are studying classes specified by oracle machines, we must also consider operations on the oracle set.

The first result is a Representation Lemma that shows how a language accepted by a time-bounded oracle machine can be represented algebraically. This result is used to provide characterizations of the various classes studied, e.g., the class NP(A) of languages accepted in polynomial time by nondeterministic oracle machines using A as an oracle.

The characterizations established here show that many classes that are apparently quite different have very similar algebraic structures. Further, they show that the class of regular sets can be used as a "sub-basis" for the generation of a wide variety of classes of languages specified by oracle machines while using quite simple operations on languages.

Proofs of the results presented here may be found in [5].

SECTION 1. PRELIMINARIES

It is assumed that the reader is familiar with the basic concepts from the theories of automata, computability, and formal languages. Some of the concepts that are most important for this paper are reviewed here and notation is established.

For a string w, $|w|$ denotes the length of w; e denotes the empty string and $|e| = 0$; if a is a symbol and y is a string, then $|ay| = 1 + |y|$.

The reversal w^R of a string w is the string obtained by writing w in reverse order: $e^R = e$ and if a is a symbol and y is a string then $(ay)^R = y^R a$.

For a language A, let Σ be the smallest (finite) alphabet such that $A \subseteq \Sigma^*$ and define $\bar{A} = \Sigma^* - A$ and $A \oplus \bar{A} = cA \cup d\bar{A}$ for any two symbols $c, d \notin \Sigma$.

[†] This research was supported in part by the National Science Foundation under Grants MCS77-11360 and DCR-74-15091.

[††] While this work was done, this author was at the Department of System Science, University of California at Los Angeles.

For a class $\mathcal{C}$ of languages, let co-$\mathcal{C}$ be the class of complements of languages in $\mathcal{C}$, i.e., if $L \in \mathcal{C}$ and $L \subseteq \Gamma^*$ for some finite alphabet Γ, then $\Gamma^* - L \in$ co-$\mathcal{C}$.

Recall that a homomorphism (between free monoids) is a function $h : \Sigma^* \rightarrow \Delta^*$ such that for all $x, y \in \Sigma^*$, $h(xy) = h(x)h(y)$. A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is nonerasing if $|w| > 0$ implies $|h(w)| > 0$. A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is linear-erasing on language $L \subseteq \Sigma^*$ if there is a constant $k > 0$ such that for all $w \in L$ with $|w| \geq k$, $|w| \leq k|h(w)|$, and is polynomial-erasing on language $L \subseteq \Sigma^*$ if there is a constant $k > 0$ such that for all $w \in L$ with $|w| \leq k$, $|w| \leq |h(w)|^k$. More generally, let $\mathcal{F}$ be a class of nondecreasing functions such that if $f \in \mathcal{F}$, then $\lim \frac{f(n)}{n} \geq 0$. A homomorphism $h : \Sigma^* \rightarrow \Delta^*$ is $\mathcal{F}$-erasing on language $L \subseteq \Sigma^*$ if there is a constant $k > 0$ and a function $f \in \mathcal{F}$ such that for all $w \in L$ with $|w| \geq k$, $|w| \leq kf(|h(w)|)$.

A class $\mathcal{C}$ of languages is closed under (nonerasing, linear-erasing) homomorphism if for every language $L \in \mathcal{C}$ and every homomorphism h (that is nonerasing, linear-erasing on L), the language $h(L) = \{h(w) \mid w \in L\}$ is in $\mathcal{C}$. More generally, let $\mathcal{F}$ be a class of nondecreasing functions such that if $f \in \mathcal{F}$, then $\lim \frac{f(n)}{n} \geq 0$. A class $\mathcal{C}$ of languages is closed under $\mathcal{F}$-erasing if for every language $L \in \mathcal{C}$ and every homomorphism that is $\mathcal{F}$-erasing on L, $h(L)$ is in $\mathcal{C}$.

Let n be a positive integer and let ρ be a function from $\{1,\ldots,n\}$ to $\{1,R\}$. Let L be a language and let $h_1,\ldots,h_n$ be homomorphisms. The language $<\rho;h_1,\ldots,h_n>(L) = \{h_1(w)^{\rho(1)}\ldots h_n(w)^{\rho(n)} \mid w \in L\}$ is a homomorphic replication of type ρ on L. A class $\mathcal{C}$ of languages is closed under homomorphic replication if for every $n > 0$, every function $\rho : \{1,\ldots,n\} \rightarrow \{1,R\}$, every language $L \in \mathcal{C}$, and every n homomorphisms $h_1,\ldots,h_n$, the language $<\rho;h_1,\ldots,h_n>(L)$ is in $\mathcal{C}$.

Let $\mathcal{C}$ be a class of languages and let $\mathcal{F}$ be a class of nondecreasing functions such that if $f \in \mathcal{F}$, then $\lim \frac{f(n)}{n} \geq 0$. If for every $n > 0$, every function $\rho : \{1,\ldots,n\} \rightarrow \{1,R\}$, every language $L \in \mathcal{C}$, and every n homomorphisms $h_1,\ldots,h_n$, each of which is nonerasing ($\mathcal{F}$-erasing on L), the language $<\rho;h_1,\ldots,h_n>(L)$ is in $\mathcal{C}$, then $\mathcal{C}$ is closed under nonerasing ($\mathcal{F}$-erasing) homomorphic replication.

The operation of homomorphic replication has been used recently [3,4, 6,7] to characterize classes of languages arising in the study of automata, computability, and computational complexity. The basis for much of this use is the fact that if $\rho : \{1,2\} \rightarrow \{1,R\}$ is defined by $\rho(1) = 1$ and $\rho(2) = R$, then the class of all languages of the form $<\rho;h_1,h_2>(L)$ for arbitrary homomorphisms h_1 and h_2 and an arbitrary regular set L is the class of all linear context-free languages.

SECTION 2. THE REPRESENTATION LEMMA

In this section we define "oracle machines" and provide a basic Representation Lemma for languages specified by time-bounded oracle machines.

An *oracle machine* is a multitape Turing machine M with a distinguished work tape, the *query* tape, and three distinguished states, $q_?$, q_{yes}, and q_{no}. At some step of a computation on an input string w, M may transfer into the state $q_?$. In state $q_?$ M transfers into the state q_{yes} if the string currently appearing on the query tape is in some *oracle set* A; otherwise, M transfers into the state q_{no}; in either case, the oracle tape is instantly erased. The set of strings *accepted by* M *relative to the oracle set* A is $L(M,A) = \{w \mid$ there is an accepting computation of M on input w when the oracle set is $A\}$.

Representation Lemma. Let M be an oracle machine that runs in time t(n) and has tape alphabet Δ. There exist homomorphisms h_1 and h_2 and a language L_M such that (i) for any oracle set $A \subseteq \Delta^*$, $L(M,A) = h_1(L_M \cap h_2^{-1}((A \oplus (\Delta^*-A))^*))$, (ii) L_M is accepted in linear time by a deterministic multitape Turing machine, and (iii) for all $w \in L_M$, $|w| \leq t(|h_1(w)|)$.

SECTION 3. MAIN RESULTS

Let $\mathcal{F}$ be a class of nondecreasing functions such that if $f \in \mathcal{F}$, then $\lim f(n)/n^2 \geq 0$ and such that $\mathcal{F}$ is closed under composition.

We shall state our results for classes of languages specified by nondeterministic oracle machines whose running times are bounded by functions from a class that has the properties that we require of $\mathcal{F}$. For example, the class of polynomial running times that grow at least as fast as n^2 is a class of functions that can be used to specify the class NP(A) of languages that are nondeterministic polynomial time in A.

Theorem 1. For any language A. the class of languages accepted by nondeterministic oracle machines that use A as the oracle set and that operate within time bounds in the class $\mathcal{F}$ is the smallest class of languages containing the regular sets and the language $(A \oplus \bar{A})^*$ and closed under intersection, inverse homomorphism, and $\mathcal{F}$-erasing homomorphic replication.

From this general theorem we have the following more specific characterizations.

Corollary 2. For any language A, the class of languages that are recursively enumerable in A is the smallest class of languages containing the regular sets and the language $(A \oplus \bar{A})^*$ and closed under intersection, inverse homomorphism, and (arbitrary) homomorphic replication.

Corollary 3. For any language A, the class of languages that are recursive (primitive recursive) in A is the smallest class of languages containing the regular sets and the language $(A \oplus \bar{A})^*$ and closed under intersection, inverse homomorphism, and recursive-erasing (primitive recursive-erasing) homomorphic replication.

Corollary 4. For any language A, the class NP(A) of languages that are nondeterministic polynomial time in A is the smallest class of languages containing the regular sets and the language $(A \oplus \bar{A})^*$ and closed under intersection, inverse homomorphism, and polynomial-erasing homomorphic replication.

Applying the techniques of [4], particularly those used in the proof of Theorem 2.3, to the characterization of NP(A) given in Corollary 4, we obtain the following result.

Corollary 5. For any language A, the class NP(A) of languages that are nondeterministic polynomial time in A is the class of all languages of the form $h(g^{-1}((A \oplus \bar{A})^*) \cap L_1 \cap L_2 \cap L_3)$ where L_1, L_2, and L_3 are linear context-free languages, g is a homomorphism, and h is a homomorphism that is polynomial-erasing on $g^{-1}((A \oplus \bar{A})^*) \cap L_1 \cap L_2 \cap L_3$.

Theorem 1 has a simple generalization in case the oracle sets are taken from a class of languages. Instead of stating the most general form of this result, we restrict attention to the case where it is most likely to be useful. Note that the following corollary applies to classes of the form NP(A).

Corollary 6. Suppose $\mathcal{C}$ is a class of languages that is closed under intersection with regular sets, inverse homomorphism, concatenation and Kleene *. Then $NP(\mathcal{C}) = \{NP(A) \mid A \in \mathcal{C}\}$ is the smallest class of languages containing $\mathcal{C}$ and co-$\mathcal{C}$ and closed under intersection and polynomial-erasing homomorphic replication.

We have required that the class of bounding functions used has the property that each function in the class must grow as fast as n^2. The class of linear functions does not meet these conditions and, indeed, we have not been able to characterize the class of languages that are linear time in an oracle set in the form of Theorem 1. However if we require that the class contain the deterministic context-free languages, we do have a characterization.

Proposition 7. For any language A, the class of languages accepted in linear time by nondeterministic oracle machines that use A as the oracle set is the smallest class of languages containing the deterministic context-free languages and closed under intersection, inverse homomorphism, and linear-erasing homomorphic replication.

The class of extended rudimentary relations is the smallest class of string relations containing the concatenation relation (concat(x,y,z) is true if and only if xy = z) and and closed under the Boolean operations, explicit transformation, and polynomial-bounded existential quantification. In order to study the extended rudimentary relations as a class of languages, we fix an encoding of strings to languages. Let S be an alphabet and # be a symbol not in S. If R is an m-ary string relation over S, then the sequential encoding of R is $\sigma(R) = \{x_1\#\ldots x_m\# \mid (x_1,\ldots,x_m) \in R\}$. The class EXRUD of extended rudimentary languages is defined to be the class of language encodings of the extended rudimentary relations: EXRUD = $\{\sigma(R) \mid$ R is an extended rudimentary relation$\}$. For any language A, the class EXRUD(A) of languages that are extended rudimentary in A is defined to be the class of language encodings of the smallest class of string relations containing the concatenation relation and the relation representing membership in A, and closed under the Boolean operations, explicit transformation, and polynomial-bounded existential quantification (see [11,14]).

We shall characterize the class EXRUD(A) in terms of the regular sets, the language A, and certain operations on languages.

Let A be a language. Let $\Sigma_1^p(A)$ be NP(A). For each $i \geq 1$, let $\Pi_i^p(A)$ be the class of complements of languages in $\Sigma_i^p(A)$, so that $\Pi_i^p(A) = \text{co-}\Sigma_i^p(A)$, and let $\Sigma_{i+1}^p(A)$ be the image of $\Pi_i^p(A)$ under polynomial-erasing homomorphism, $\Sigma_{i+1}^p(A) = \{h(L) \mid L \in \Pi_i^p(A)$ and h is a homomorphism that is polynomial-erasing on L$\}$. The structure $\Sigma_1^p(A) \subseteq \Sigma_2^p(A) \subseteq \ldots$ is known as the polynomial hierarchy relative to A (see [1,2,8,10,11,14]). The class EXRUD(A) has been shown [14] to be equal to the union of this hierarchy, EXRUD(A) = $\bigcup_{i\geq 1} \Sigma_i^p(A)$. In addition, it is shown in [11,14] that since $\Sigma_1^p(A)$ = NP(A), for each $i \geq 1$, $\Sigma_{i+1}^p(A) = \text{NP}(\Sigma_i^p(A))$. These facts are very useful in establishing our characterizations of EXRUD(A).

Theorem 8. For any language A, the class EXRUD(A) of languages that are extended rudimentary in A is the smallest class of languages containing the regular sets and the language A and closed under intersection, complementation, inverse homomorphism, and polynomial-erasing homomorphic replication.

It is easy to see that for any language A, NP(A) is closed under complementation if and only if NP(A) = EXRUD(A). It is shown in [1] that that there exists a language A such that NP(A) is not closed under complementation and there exists a language B such that NP(B) is closed under complementation. At this time it is not known whether there exists a language C such that the hierarchy $\Sigma_1^p(C) \subseteq \Sigma_2^p(C) \subseteq \ldots$ is in fact an infinite hierarchy, that is, whether for all $i \geq 1$, $\Sigma_i^p(C) \neq \Sigma_{i+1}^p(C)$, although partial results are known [2]. The most interesting case is that of NP = NP($\emptyset$); it is not known whether NP is closed under complementation and it is not known whether the polynomial hierarchy $\Sigma_1^p(\emptyset) \subseteq \Sigma_2^p(\emptyset) \subseteq \ldots$ is in fact an infinite hierarchy [10,12].

Results similar to Theorem 8 are the following.

Corollary 9. For any language A the class of languages that are arithmetical in A is the smallest class of languages containing the regular sets and the language A and closed under the Boolean operations, inverse homomorphism, and (arbitrary) homomorphic replication.

Corollary 10. For any language A the class of languages that are rudimentary in A is the smallest class of languages containing the regular sets and the language A and closed under the Boolean operations, inverse homomorphism, and linear-erasing homomorphic replication.

REFERENCES

1. T. Baker, J. Gill and R. Solovay, Relativizations of the P = ? NP question, SIAM J. Comput. 4 (1975), 431-442.
2. T. Baker and A. Selman, A second step towards the polynomial hierarchy, Theoret. Comput. Sci., to appear.
3. R. Book, Simple representations of certain classes of languages, J. Assoc. Comput. Mach. 25 (1978), 23-31.
4. R. Book and M. Nivat, Linear languages and the intersection closures of classes of languages, SIAM J. Comput. 7 (1978), 167-177.
5. R. Book and C. Wrathall, On languages specified by relative acceptance, Theoret. Comput. Sci., to appear.
6. S. Greibach, One-way finite visit automata, Theoret. Comput. Sci. 6 (1978), 175-221.
7. K. Klingenstein, ρ-matrix languages, Theoret. Comput. Sci., to appear.
8. R. Ladner, On the structure of polynomial time reducibility, J. Assoc. Comput. Mach. 22 (1975), 155-171.
9. R. Ladner, N. Lynch and A. Selman, A comparison of polynomial time reducibilities, Theoret. Comput. Sci. 1 (1975), 103-123.
10. L. Stockmeyer, The polynomial-time hierarchy, Theoret. Comput. Sci. 3 (1976), 1-22.
11. C. Wrathall, Subrecursive Predicates and Automata, Ph.D. dissertation, Harvard University, 1975.
12. C. Wrathall, Complete sets and the polynomial-time hierarchy, Theoret. Comput. Sci. 3 (1976), 23-33.
13. C. Wrathall, Rudimentary predicates and relative computation, SIAM J. Comput. 7 (1978), 194-209.
14. C. Wrathall, The linear and polynomial-time hierarchy, to appear.

Real-time Simulation of Multihead Reset Tape Units[†]
Preliminary Version

S. Rao Kosaraju
Electrical Engineering Department
The Johns Hopkins University
Baltimore, MD 21218

Abstract

It is proved that any multihead reset Turing machine can be simulated in real-time by a multitape Turing machine - which settles a well-known open problem. It is also established that any n-head reset Turing machine can be simulated in real-time by a $\binom{n}{2}$-tape 2-head reset Turing machine - which resolves another open problem.

I. Introduction

It is known [FMR,LS] that any multihead Turing machine can be simulated in real-time by a multitape Turing machine. The proof of [FMR] is based on a very interesting tape folding technique. The recent proof of [LS] does not involve any tape folding; it hinges on a clever idea of information updating on linear tapes.

It is also known [FR,LS] that any single head reset Turing machine can be simulated in real-time by a multitape Turing machine. The proofs of both [FR] and [LS] rely on first simulating a reset Turing machine by an infinite set of double-ended queues (deques). Leong, Seiferas [LS] first establish that any stack of deques can be simulated in real-time by a multitape Turing machine, and then demonstrate that the reset machine can be simulated in real-time by a fixed number of stacks of deques. Here we show that any multihead reset Turing machine can also be simulated in real-time by a fixed number of stacks of deques.

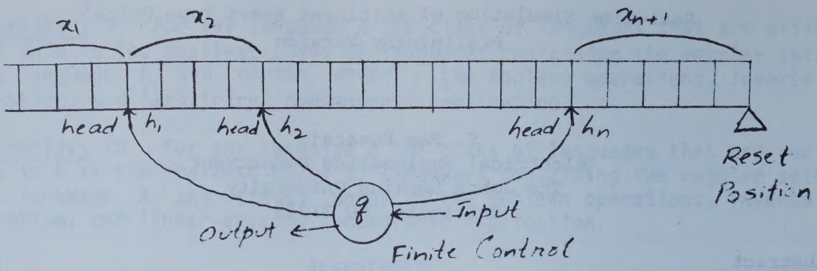
Section II introduces the multihead reset Turing machine model and the stack of deques model. Section III gives a real-time simulation of any n-head reset Turing machine by a $\binom{n}{2}$-tape 2-head reset Turing machine. Finally, it is established in Section IV that a 2-head reset Turing machine can be simulated in real-time by a multitape Turing machine.

For ease of presentation and understanding, the treatment will be done at an informal level. In addition, a knowledge of the Turing machines and pushdowns is assumed. Section II does not go into all the details of these models. This section is meant only to introduce the conventions followed in the paper. Some of the conventions are not spelled out in detail. Even though the fine details of the construction would depend on the particular assumptions made, the basic ideas will not change.

II. Basic Models

An n-head reset Turing machine (we abbreviate Turing machine by TM) has n heads on a tape which is infinite to the left as shown below.

[†] Supported by NSF Grant No. MCS 75-09904



At any instant, every head resides in between 2 adjacent squares[†]. Initially every head is at the reset position, and every square contains a blank, '$\not{b}$'. The operations of the heads are controlled by a central finite state control. At any instant, the finite state control accesses the contents of the two squares adjacent to each head (for the reset position only the left square) and receives an external input symbol. Based on this, in a single step, it changes the contents of the squares accessed, moves the heads independently left or right by one square or resets to the reset position, and may output some symbol. A head cannot move to the right of the reset position, and the blank symbol cannot be printed. At the instant shown in the figure, the tape content $x_1x_2\ldots x_{n+1}$ segmented into $(x_1,x_2,\ldots,x_{n+1})$ gives the tape configuration, and $(h_1,h_2,\ldots,h_n)$ gives the (left to right) ordering of the heads on the tape.[††] The tape configuration with the head ordering can also be represented by $x_1h_1x_2h_2\ldots h_nx_{n+1}$. This together with the state of the finite state control specifies the complete configuration.

A t-tape n-head reset TM has t tapes, each having n heads and operating as above, all the nt heads being controlled by a central finite state control. When the heads of the TM cannot reset, the resulting model is a t-tape n-head TM. A t-tape 1-head TM is simply denoted as a t-tape TM.

In a t-tape n-head TM, if any head on a tape can jump to the position of any other head on that tape in a single step, then the resulting model is a t-tape n-head jump TM. In a t-tape (n+1)-head TM, if one head on each tape is designated as a reset head and any head on a tape can jump to the reset head on that tape in a single step, then the resulting model is a t-tape n-head movable tab TM.

A deque (double ended queue) is a linear string of symbols, taken from a finite alphabet, in which the access is limited to the two ends. In a single step, a symbol can be deleted or appended to either end. A d-depth stack of deques (d-depth stadeq) is a stack of deques - that is, every level ("stack symbol") is a deque, and the access is limited to the top d deques. Any deque operation can be performed on an accessible deque. Deletion (pop) or creation (push) of a deque can be performed only at the top of the stack. The deleted or created deque must be empty. The operation is controlled by a finite state control, which at any instant accesses the ends of the top d deques. It can also test whether an accessible deque is empty, or is of length less than or equal to a specified constant. When t d-depth stadeqs are controlled by a central finite state control, the resulting model is a t unit d-depth stadeq. At any instant, let the stadeq contain $y_1, y_2, \ldots, y_m$, where each y_i is a deque and y_1 and y_m are the bottom and the top deques, respectively. For $i=1,\ldots,m-1$, denote the boundary between y_i and y_{i+1} as seam i. For $k \geq 1$, seams $m-1,\ldots,m-k$ form the top k seams.

[†] Most of the literature assumes that a head resides on a square, but the results remain invariant under this change.

[††] For only $i \geq 0$, $\not{b}^i x_1 x_2 \ldots x_{n+1}$ also denotes the same tape content.

An automaton M' is said to simulate in real-time (RT simulate) an automaton M if and only if M' requires one step for every step of M, and M and M' are functionally (input-output relationship) equivalent.

An automaton M' is said to k-RT simulate an automaton M if and only if M' requires at most k steps for every step of M, and M and M' are functionally equivalent.

We now state some well-known results without proofs.

Lemma 2.1 [LS,S]: Any d-depth stadeq M can be RT simulated by a d unit 1-depth stadeq.

The proof is based on uniformly distributing the deques of M into d stadeqs such that at every instant the top d deques of M form the top deques of the d stadeqs.

Lemma 2.2 [HS]: If there exists a k such that an automaton M can be k-RT simulated by a multitape TM M', then M can be RT simulated by a multitape TM.

Lemma 2.3 [LS]: Any t unit 1-depth stadeq can be RT simulated by an 8t-tape TM.

Lemma 2.4 [LS]: There exists a k such that any 1-head reset TM can be k-RT simulated by a 3-depth stadeq.

As a consequence of Lemmas 1-4, we have:

Lemma 2.5 [LS]: Any 1-head reset TM can be RT simulated by a 24-tape TM.

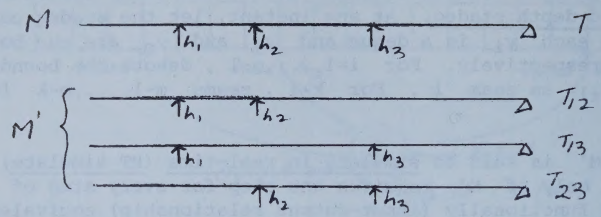
III. Simulation of n head reset Turing machine by 2 head reset Turing Machine

In this section and the next, we establish that any n head reset Turing machine can be RT-simulated by a multitape Turing machine. This goal is achieved by first showing in this section that the n head problem can be "reduced" to a 2 head problem.

Leong and Seiferas [LS] have shown that any 1 tape n head reset Turing machine (TM) can be RT simulated by cn^3 tape 2 head reset TM, for some constant $c>0$ [†]. They pose the open problem whether the RT simulation can be done with only $O(n^2)$ tape 2 head reset TM. Here we demonstrate such a simulation. The key to the procedure is the following RT simulation of any 1 tape 3 head reset TM by a 3 tape 2 head reset TM.

[†] They discuss the simulation for multidimensional TM model. Our procedure applies verbatim to their problem.

3.1 Reduction of 3 heads to 2 heads



The 3 head TM M is replaced by a 3 tape 2 head TM M' as shown above. Each T_{ij} tape, $i,j \varepsilon \{1,2,3\}$ and $i \neq j$, contains heads h_i and h_j. The positions of the heads of M' correspond to the positions of the heads of M. When any head h_i of M prints a symbol, the corresponding head h_i on two of the 3 tapes of M' could print the same symbol; however the information on the remaining tape cannot be updated. Subsequently, when a different head visits that square, M' should be able to find the latest symbol written in that square. To facilitate this, every head of M' prints symbols of the form (Z,k) where Z is the symbol printed by the corresponding head of M and $k \varepsilon \{0,1,2\}$. Initially, every square of T_{ij} is assumed to contain $(-,k)$, where '-' is the blank symbol of M and $k=0$ for the tape T_{23} and $k=1$ for the other two tapes.

Simulation Algorithm

At any instant and for any $i \varepsilon \{1,2,3\}$, let the two h_i heads of M' scan (Z_1,j_1) and (Z_2,j_2). At that instant, let the correct symbol scanned by the h_i head of M be Z. Also let the h_i head of M change Z to Z' in the next step. We now describe how M' finds Z from (Z_1,j_1) and (Z_2,j_2), and what M' prints in response.

(i) If $j_1 = j_2$, then $Z=Z_1=Z_2$ and each h_i head of M' prints (Z',j_1).

(ii) Otherwise, find the $k \varepsilon \{1,2\}$ that satisfies $j_k = j_{3-k}+1 \pmod 3$. Then $Z=Z_k$ and each h_i head of M' prints (Z', j_k'), where $j_k' = j_k+1 \pmod 3$.

The case of more than one head scanning the same square of M is treated as though only one of them scans that square. The correctness of this procedure follows from the following lemma:

Lemma 3.1: At any instant for any square m of M containing Z, if the three corresponding squares of M' contain (Z_1,j_1), (Z_2,j_2) and (Z_3,j_3), then

(a) $\#\{j_1,j_2,j_3\}=2$,

(b) if $j_{k_1} = j_{k_2}$, $k_1 \neq k_2$, then $Z = Z_{k_1} = Z_{k_2}$, and $j_{k_1} = j_{k_3}+1 \pmod 3$, where $\{k_1,k_2,k_3\} = \{1,2,3\}$.

Proof: We prove this by induction on the step number. Initially, i.e. before step 1, the 3 squares corresponding to any square of M contain $(-,0)$, $(-,1)$ and $(-,1)$, which satisfy (a) and (b). Assume that (a) and (b) hold after t steps. If no head is on square m, then the next step will not disturb (a) and (b). Without loss of generality, assume that head h_1 is on square m and changes Z to Z' in the next step. Before this change, let the h_1 heads of M' scan (Z_1,j_1) and (Z_2,j_2), and let the content of the square m of tape T_{23} be (Z_3,j_3).

Case 1: $j_1=j_2$

By the inductive hypothesis, $j_1=j_3+1 \pmod 3$ and $Z=Z_1=Z_2$; and by (i) of the algorithm the two h_1 heads print (Z',j_1). The three corresponding symbols after printing will be (Z',j_1), (Z',j_2) and (Z_3,j_3) where $j_1=j_2=j_3+1 \pmod 3$. Verify that (a) and (b) hold for these symbols.

Case 2: $j_1 \neq j_2$.

By the inductive hypothesis, $j_3 \varepsilon \{j_1,j_2\}$. Without loss of generality, assume that $j_3=j_1$. Then $j_1=j_2+1 \pmod 3$. In step (ii) of the algorithm, k has value 1, Z is taken as Z_1 - the correct symbol, and the h_1 heads print (Z', j_1'), where $j_1'=j_1+1 \pmod 3$. After this step, the three corresponding symbols are (Z', j_1'), (Z',j_1') and (Z_3,j_1). Note that $\#\{j_1',j_1',j_1\}=2$ - satisfying (a), and $j_1'=j_1+1 \pmod 3$ and Z' is the correct symbol - satisfying (b).

Q.E.D.

This simulation algorithm relies on the fact that just by knowing the contents of 2 of the 3 squares corresponding to a square of M, M' can find out the latest head that visited that square and the correct symbol in that square. If head h_i scans (Z_1,j_1) and (Z_2,j_2) on tapes[†] T_{ii_1} and T_{ii_2}, respectively, where $\#\{i,i_1,i_2\}=3$, then

(i) if $j_1=j_2$ then h_i has visited that square last and printed Z_1 $(=Z_2)$, and

(ii) if $j_k=j_{3-k}+1 \pmod 3$, $k=1$ or 2, then h_{i_k} has visited that square last and printed Z_k.

The validity of this observation can be easily seen from the above lemma.

Reduction of n heads to 2 heads

Theorem 3.1: Any 1 tape n head reset TM M can be RT simulated by a $\binom{n}{2}$ tape 2 head reset TM M'.

Proof: If the n heads of M are $h_1,h_2,\ldots,h_n$, then for $1 \leq i,j \leq n$ and $i \neq j$, tape T_{ij} of M' contains the heads h_i and h_j and their positions match with the positions of the corresponding heads of M. Any square m of the tape T_{ij} contins $(Z,(a_{ijm1},a_{ijm2}\ldots,a_{ijmn}))$, where Z is a symbol of M and $(a_{ijm1},\ldots,a_{ijmn})$ is an n-tuple from $\{0,1,2\}^n$, with the constraint $a_{ijmi}=a_{ijmj}=0$[††]. For any $k \notin \{i,j\}$, the a_{ijmk} performs the same function as the additional number printed in the square m of the tape T_{ij} while reducing the 3 heads h_i,h_j and h_k to the 3 tape 2 head model.

At any instant when some head h_i scans some square m, M' can find out the latest head that visited square m and the symbol it printed as follows. For any j and k, with $\#\{i,j,k\}=3$, making use of a_{ijmk} on tape T_{ij} and a_{ikmj} on tape T_{ik}, M' can find the latest head out of h_i,h_j and h_k that visited square m. By varying j and k, M' can find the latest head, say h_p, out of $h_1,\ldots,h_n$ that visited square m. The latest symbol printed in square m can be obtained from the symbol scanned by h_i on the tape T_{ip} . The remaining details can be easily worked out.

[†] T_{ij} and T_{ji} refer to the same tape.

[††] Since a_{ijmi} and a_{ijmj} are fixed as 0, we can replace the n-tuples by (n-2)-tuples. However the description becomes simpler with n-tuples.

The same procedure can be applied to simulate n processors sharing a common variable, which can be accessed by at most one processor at a time, by n processors in which not more than 2 processors can share a variable (Figure 3.1).

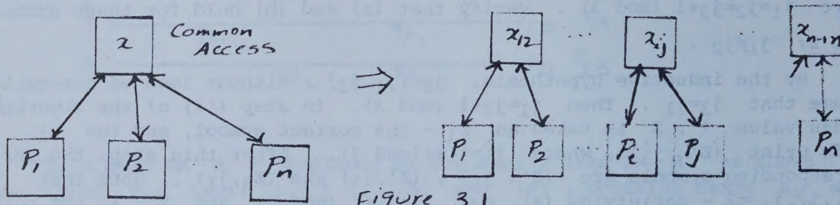


Figure 3.1

IV. Simulation of 2-head reset Turing machines

In this section, we prove that any 2 head reset TM can be RT simulated by a multitape TM. First we introduce a generalized stadeq.

A (k,r) stadeq is a stack of deques in which at any instant the access is limited to the top k deques and at most r other deques taken from the set of accessible deques at the immediately preceding instant.

Theorem 4.1: For any $k \geq 2$ and $r \geq 0$, any (k,r) stadeq M can be RT simulated by a (k+r) unit 1-depth stadeq M' .

Proof: The deques in the stack of M get distributed among the (k+r) stacks of M' such that at any instant all the accessible deques of M form the top deques of M' . Every deque of M' is a pair like (y,i) where y is a deque of M and $i \in \{1,2,\ldots,k+r\}$ - (y,i) signifies that the deque immediately below y in M is stored in the i^{th} stack of M' . Since a deque is a string of symbols, but not an ordered pair, in actual implementation, the i has to be incorporated within the alphabet structure of the deque by expanding its alphabet. The minor problem of the case of the empty deque can be handled by introducing a deque symbol to designate an empty deque. The finite control of M' keeps track of the index of the stack which contains each accessible deque of M .

Now we investigate how M' handles the operations of M . If M appends or deletes symbols from an accessible deque, M' can reproduce it since each accessible deque of M is the top deque of a stack of M' . The operation of M' when M pushes or pops a deque is given below. (Let I_{t-j+1} be the index of the stack of M' that contains the j^{th} topmost deque of M , for $j=1,\ldots,k$ (I_t for the topmost deque)) .

Case 1: M pushes a new deque, y .

Just before this operation if M can access less than k+r deques, then the top deque of one of the stacks of M' did not contain an accessible deque. Then M' pushes (y,I_t) on that stack. Otherwise, one of the previously accessible (k+r) deques of M becomes inaccessible after y gets pushed. M' pushes (y,I_t) on the stack that contains the inaccessible deque.

Case 2: M pops the top deque.

M' pops the top deque of the I_t-th stack and the indices of the stacks containing the top k-1 deques of M are $I_{t-1},\ldots,I_{t-k+1}$. If the top entry of I_{t-k+1} is (z,j) then the k^{th} topmost deque of M is the top deque of the j^{th} stack of M' .

The remaining details can be easily provided. It is very important to observe that even when M does a sequence of pops, the top k deques of M remain as the top deques of the stacks of M' .

Theorem 4.2: Any 2-head reset TM M can be k-RT simulated by some (5,2) stadeq M' for some fixed k .

Proof: Without loss of generality, we assume that the 2 heads of M cannot cross each other making nonreset moves; i.e. the head ordering remains invariant when the heads move left or right. We can also assume that only one head moves in a single step. During the simulation the following relationship is maintained between the tape configuration of M and the contents of the deques of M' .

At some instant, let the stadeq of M' contain $y_1,y_2,\ldots,y_m$, where each y_i is a deque and y_m is the topmost deque. At the same instant, the tape content of M must be $y_1y_2\ldots y_m$, and the tape configuration must be $(y_1\ldots y_{s_1},y_{s_1+1}\ldots y_{s_2},y_{s_2+1}\ldots y_m)$ where $1\le s_1<s_2$ and $s_2\varepsilon\{m-2,m-1\}$. In addition, the following conditions will be satisfied.[†]

$$|y_j| > \sum_{i=0}^{m-j-3} 2^i|y_{j+3+i}| \text{ , for } j = 1,2,\ldots,m-3 .$$

Now we show that M' can simulate a step of M by at most k steps, for some fixed k , maintaining all the conditions specified above.

We simplify the details by assuming that the heads of the TM are on the stadeq itself. For example, in the above correspondence where the tape configuration is $(y_1\ldots y_{s_1},y_{s_1+1}\ldots y_{s_2},y_{s_2+1}\ldots y_m)$ we say that the two heads are at the s_1th and the s_2th seams.

Initially M' sets up 3 empty deques, y_1,y_2,y_3 , with heads h_1 and h_2 at seams 1 and 2 , respectively: $y_1h_1y_2h_2y_3$, $y_1=y_2=y_3=\Lambda$ (empty).

At some instant let M' have m deques $y_1,y_2,\ldots,y_m$, with heads h_1 and h_2 at seams s_1 and s_2 such that $s_1<s_2$. Note that the corresponding tape configuration of M is $(y_1\ldots y_{s_1},y_{s_1+1}\ldots y_{s_2},y_{s_2+1}\ldots y_m)$. If we transfer the leftmost symbol of y_{s_1} to the right of y_{s_1-1} , we get another stadeq representation for the same tape configuration of M . In general, we can transfer a symbol from any deque to an adjacent deque when there is no head in between the 2 deques. Let $y_i \xrightarrow{c} y_{i+1}$ denote the transfer of c symbols, one at a time, from the right of y_i to the left of y_{i+1} , and let $y_{i+1} \xrightarrow{c} y_i$ denote similar transfer from the left of y_{i+1} to the right of y_i . Thus $y_i \xrightarrow{c} y_{i+1}$ or $y_{i+1} \xrightarrow{c} y_i$ requires c steps of the stadeq. Let $*y_i \xrightarrow{c} y_{i+1}$ be defined as follows:

If $|y_i| \geq c$, then $y_i \xrightarrow{c} y_{i+1}$, otherwise (let $|y_i|=a$)

$$*y_{i-1} \xrightarrow{c-a} y_i \text{ followed by}$$

$$y_i \xrightarrow{c} y_{i+1} .$$

Similarly $*y_{i+1} \xrightarrow{c} y_i$ can be defined.

[†] $|x|$ denotes the length of x .

The next step of M can be a left move, a right move, or a reset move.

Case 1: Let head h_i, i=1 or 2, move right by one square. Then M' performs $*y_{s_i+1} \xrightarrow{1} y_{s_i}$ and if $s_i<m-4$, M' calls the procedure Improve$(y_1,y_2,\dots,y_m,s_1,s_2)$ given below.

Case 2: Let head h_i, i=1 or 2, move left by one square. Then M' performs $*y_{s_i} \xrightarrow{1} y_{s_i+1}$ and then calls Improve$(y_1,y_2,\dots,y_m,s_1,s_2)$.

Case 3: If h_i resets, then M' pushes an empty deque y_{m+1} $(=\Lambda)$ on top, sets $s_i=m$ and then calls Improve$(y_1,y_2,\dots,y_{m+1},s_{3-i},s_i)$.

The Improve procedure is as follows:

Improve$(y_1,\dots,y_m,s_1,s_2)$

If $m\leq 3$, no operation is necessary - simply return.
If $m=4$, concatenate 2 blank symbols of M to the left of y_1 and return.
In the case $m>4$, the operations are given in the Table 4.1 (without loss of generality, we assume that $s_1<s_2$ and $s_2 \in \{m-2,m-1\}$) .

It can be shown that this simulation is a k-RT simulation, for some fixed k .

Positions of heads	Conditions Satisfied	Operations to be performed
I. $s_2 \neq m-1$ (no head at the m-1th seam)	$\lvert y_m\rvert \geq 1$	$y_m \xrightarrow{1} y_{m-1}$.
	$\lvert y_m\rvert = 0$	Pop y_m , call Improve$(y_1,\dots,y_{m-1},s_1,s_2)$.
II. $s_2=m-1, s_1=m-2$	$\lvert y_{m-2}\rvert \geq 3$	$y_{m-2} \xrightarrow{3} y_{m-3}$, then $y_{m-3} \xrightarrow{2} y_{m-4}$.
	$\lvert y_{m-2}\rvert < 3$	$y_{m-2} \xrightarrow{a} y_{m-3}$, let $s_1=m-3$,
	(let $\lvert y_{m-2}\rvert=a$)	and go to Case III.
III. $s_2=m-1, s_1=m-3$	$\lvert y_{m-1}\rvert \geq 3$	$y_{m-4} \xrightarrow{1} y_{m-3}$ and $y_{m-1} \xrightarrow{3} y_{m-2}$.
	$\lvert y_{m-1}\rvert < 3$	$y_{m-1} \xrightarrow{a} y_{m-2}$, let $s_2=m-2$,
	(let $\lvert y_{m-1}\rvert=a$)	and go to Case I.
IV. $s_2=m-1, s_1=m-\ell, \ell\geq 4$	$\lvert y_{m-1}\rvert \geq 2$	$y_{m-1} \xrightarrow{2} y_{m-2}$, then $y_{m-2} \xrightarrow{2} y_{m-3}$.
	$\lvert y_{m-1}\rvert < 2$	$y_{m-1} \xrightarrow{a} y_{m-2}$, let $s_2=m-2$,
	(let $\lvert y_{m-1}\rvert=a$)	and go to Case I.

Table 4.1

As a consequence of Lemmas 2.2 and 2.3 and Theorems 3.1, 4.1 and 4.2 we have:

Theorem 4.3: Any n-head reset TM can be RT simulated by a $56\binom{n}{2}$-tape TM.

It can also be shown that any n-head movable tab TM can be RT simulated by a multitape TM. However the following problem is still open.

Open problem [SV]: Can every n-head jump TM be RT simulated by a multitape TM?

References

[B] V. L. Bennison: Saving tapes in the simulation of multihead Turing machines, SIGACT NEWS, Vol. 6, No. 2, 1974, 23-26.

[FR] M. J. Fischer and A. L. Rosenberg: Limited random access machines (Preliminary Version), Proc. of Ninth Annual Symp. on Switching and Automata Theory, 1968, 356-367.

[FMR] P. C. Fischer, A. R. Meyer and A. L. Rosenberg: Real-time simulation of multihead tape units, JACM, 1972, 590-607.

[HS] J. Hartmanis and R. E. Stearns: On the computational complexity of algorithms, Trans. of the American Math. Soc., Vol. 117, 1965, 285-306.

[LS] B. Leong and J. Seiferas: New real-time simulations of multihead tape units, Proc. of the Ninth Annual ACM Symp. on Theory of Computing, 1977, 239-248.

[SV] W. J. Savitch and P. M. B. Vitányi: Linear time simulation of multihead Turing machines with head-to-head jumps, Lecture Notes in Computer Science, (ed.) G. Goos and J. Hartmanis, Vol. 52, 1977, 453-464.

[S] J. Seiferas: Personal communication.

[ST] H. J. Stoss: k-Band Simulation von k-Kopf-Turing Maschinen, Computing, Vol. 6, 1970, 309-317.

[YR] A. C. Yao and R. L. Rivest: k+1 heads are better than k , JACM, Vol. 25, 1978, 337-340.

INTERMEDIATE COMPLEXITY IN DATA MATRIX PROBLEMS

FREDERICK N. SPRINGSTEEL
Computer Science Dept., University of Missouri-Columbia 65211

(with P. P. PUDLAK
Czechoslovak Academy of Sciences, Prague)

SHORT SUMMARY: This paper examines a series of connected, concrete restrictions of NP-complete problems, based in hypothesis formation for data matrices, which reveal the following range of observed complexities (time/space simultaneous bounds): quadratic time/log space; quartic time/log space; DLOG-hard; NLOG-complete; P-time or $\log^2$ space; and (the set "D_{par}") subexponential time/$\log^2$ space, deterministically.

Many researchers have discovered new NP-complete problems (cf. [2] [8], [9] for example), while others have unearthed problems complete for finer reducibilities inside the class P (e.g., [5], [6], [7]). Sometimes the latter problems are related to analogous NP-complete problems - for example, UNIT in [6] which is P-complete and SAT, which is NP-complete [2]; more often, however, the problems on the lower levels do not relate directly to intractable problems. Most critically, few noncomplete NP problems have been "diagnosed" as non-polynomial.

Here we consider one class of problems having a single practical motivation: hypothesis formation over data bases. We find that this class alone displays a wide range of complexity within the class NP, from surprisingly P-time problems to NP-complete problems, the distinction sometimes being only whether the data is complete (always two-valued, 0 or 1) or incomplete (three-valued, including "×" for "not known"). Other problems have complexity behavior strictly between polynomial and exponential time, or between logarithmic and linear space, or both; these problems differ only by syntactic restrictions on the form of hypothesis sought. All data bases are matrix tables of two-valued properties (columns) of some objects (rows), and the hypotheses desired are (non-tautologous) disjunctions of the properties or their negations true for all objects in the data base. Offline Turing machines (or, log-cost RAMS) constitute the complexity model for making comparisons. When matrices are three-valued we evaluate disjunctions in standard three-valued logic [4]. Rows containing ×'s may fail to verify or falsify disjunctions! Sorting problems for redundant rows, with large data bases, cause storage as well as time complexity concerns. Most complexity estimates given are only proven to be upper bounds, but some optimal bounds are argued.

Notation. Let $\mathcal{Z} = \{0,1\}$ and $\mathfrak{Z} = \{0,1,\times\}$. Then $\mathcal{Z}^n$ (resp., $\mathfrak{Z}^n$) denotes the set of all n-tuple strings over {0,1} (resp., over {0,1,x}). Also, $\mathbb{2}^{m\times n}$ (resp., $\mathbb{3}^{m\times n}$) denotes the set of all m×n matrices with entries from $\mathbb{2}$ (resp., $\mathbb{3}$). The classes $\mathbb{M}_2$ and $\mathbb{M}_3$ denote the respective sets of all such finite matrices.

For $M \in \mathbb{3}^{m\times n}$, let the length of M be $\ell(M) = n$, i.e., the number of its columns. For $1 \le j \le n$, A_j denotes a j-th predicate, whose truth in M is given by the j-th column: we say "A_j is true in M," and denote this by

$M \models A_j$, iff $(\forall i \le m)\ M(i,j) = 1$. [j-th column all 1's]

Inductively one defines $M \models \phi$ for other formulae ϕ, where the universal quantifier (over the implied variable, x) is usually considered to be implicit; for example, if $d = A_{j_1} \vee \ldots \vee A_{j_k}$, we write only d instead of $(\forall x)d$. Thus,

$$M \models d \text{ iff } (\forall i)\ \{M(i,j_1) = 1 \vee \ldots \vee M(i,j_k) = 1)\}.$$

For $\alpha \in \mathbb{3}^n$, denote by d_α the (elementary) disjunction

$$\vee\{A_i : \alpha(i) = 1\} \vee \vee\{\neg A_i : \alpha(i) = 0\};$$

the length of d_α is to be $\ell(d_\alpha) = |\{i : \alpha(i) \ne x\}|$.

The set of cards of a matrix M is denoted by

$$C(M) = \{\alpha \in \mathbb{3}^n : \exists i\ \forall j\ (\alpha(j) = M(i,j))\}.$$

C(M) contains as many elements as M has distinct rows. Let + be addition in the power group $\mathbb{2}^n$; e.g. $1^n + 1^n = 0^n$. That is, all sums are boolean V (modulo 2).

The mnemonic "D", with various subscripts, denotes an (encoded) set of matrices, with possible parameters, having a true elementary disjunction of a form to be specified. For example D_{par} is the set of strings encoding pairs (M,k) where k is in $\mathbb{N}$, the natural numbers, such that some $d_\alpha, \ell(d_\alpha) \leq k$, is true in M, a two-valued matrix. "D^+" connotes positive disjunctions: we say d_α is positive iff $\alpha(i) = 1$ or x, all i, and write $\exists^+$ for "$\exists$ positive ..". We assume natural encodings; e.g., numbers in binary, m×n matrices as strings of m n-tuples of 0,1,X, etc.

I. THE TWO-VALUED CASE

Here typical results include the discovery that certain natural syntactic restrictions, such as positivity and a parametric length bound (to aid human readibility of output), make the P-time solvable set D_{max} into the NP-complete set D^+_{par} (Theorem 1). That the set D_{max}, without any restrictions, is in P (Theorem 2), requires a non-exhaustive method (dimension comparison), because for n predicates there are 2^n possible disjunctions to consider. This method partially applies to the set D_{par} in Theorem 3; the obvious exponential time algorithm can be replaced by one working in time $O(N^{\log N})$ and seemingly not in less, where $N = m \cdot n$.

Various results follow from its intermediate time/space.

Theorem 1. The (encoded) set of pairs (M,k) which have positive disjunctions of length at most k true in M:

$$D^{+}_{par} = \{(M,k) \in \mathbb{M}_2 \times \mathbb{N} : \exists^{+} d_\alpha,\ \ell(d_\alpha) \leqq k,\ M \models d_\alpha\},$$

is NP-complete.

Proof: D^{+}_{par} is in NP, by an algorithm which guesses the k-tuple d_α and proceeds to verify it. We can assume $k \leqq n$. Reduce the set $C_{par} = \{(V,E,k)$: the undirected graph (V,E) has a vertex cover of size $\leqq k\}$ to D^{+}_{par} as follows: where $V = \{1,2,\ldots,n\}$, and $|E| = m$, map (V,E,k) to (M,k) where M in $\mathbb{Z}^{m \times n}$ has non-zero entries $M(r,i) = 1 = M(r,j)$ iff (i,j) is the r^{th} edge in E. Each $S \subseteq V$ corresponds to a positive d_α where $\alpha(i) = 1$ if $i \in S$, $\alpha(i) = x$ otherwise, and conversely. So S is a vertex cover (of all edges in E) of size $\leqq k$ iff some d_α, $\ell(d_\alpha) \leqq k$, is true in (all rows of) M. Since C_{par} is an NP-complete set, so is D^{+}_{par}.

Corollary 1: The set of (encoded) matrices which have true positive disjunctions of half-maximal length:

$$D^{+}_{\frac{1}{2}} = \{M \in \mathbb{M}_2 : \exists^{+} d_\beta,\ \ell(d_\beta) \leqq \ell(M)/2, M \models d_\beta\},$$

is NP-complete.

Proof: One can reduce D^{+}_{par} to $D^{+}_{\frac{1}{2}}$. A sketch of this result appeared in [9].

Theorem 2: For any matrix $M \in \mathbb{M}_2$ it can be decided in polynomial time whether there exists an elementary disjunction d_α of maximal length, $\ell(d_\alpha) = \ell(M)$, true in M. That is, D_{max} is in P; morever, there is a polynomial time algorithm which produces such a disjunction, if one exists, using at most $\log(\cdot)$ space.

Lemma A: For $M \in \mathbb{Z}^{m \times n}$ and $\alpha \in \mathbb{Z}^n$, we have $M \models d_\alpha$ iff $\alpha^{-} \notin C(M)$, where $\alpha^{-} = \alpha + 1^n$ is the opposite of d_α.

Proof: α^{-} is the only row for which d_α is not true.

Lemma B: For $M \in \mathbb{Z}^{m\ n}$, if $C(M) \subsetneq \mathbb{Z}^n$ then an $\alpha \in \mathbb{Z}^n - C(M)$ can be found in quadratic time, and log space.

Proof: One easily implements the following algorithm, after sorting the rows [in time $(mn)^2$ and space $\log(mn)$.]

```
procedure MISSING (M,m,n):
for i = 1 until m+1 do
    1. let $\alpha$ be the i-th (ordered) row, plus 1; /store only an index i/
    2. compare $\alpha$ with each (subsequent) row in M; /use a "carry" routine/
    3. if $\alpha$ is not equal to any row in M then return $\alpha$.
```

Clearly, when $|C(M)|<2^n$ the algorithm produces a desired α; in fact it works iff $C(M) \neq \mathbb{Z}^n$. That it works in quadratic time follows from steps 1 and 2 being at most $O(mn)$. It can be made to work in $\log(mn)$ space by not actually storing α but reading it from row i, via carried bits.

Proof of Theorem 2: If M has no repeated rows, the following algorithm works in linear time (&log space) as far as deciding "$M \in D_{max}$?". Otherwise, when $m' = |C(M)|<m$, count all non-duplicate rows after sorting M (in log space), to get m' to replace m below. In this case the whole process (including sorting) is quadratic in time:

1. Compare m and 2^n, by counting the bits in binary m;
2. If $m < 2^n$, answer "Yes" [and call MISSING(M,m,n) to produce a true d_α, if desired];
3. If $m = 2^n$, answer "No".

Alternately, when $m \geqq 2^n$, one can always call MISSING to decide if $C(M) = \mathbb{Z}^n$; this takes also space $\log(mn)$ & time$(mn)^2$. In any case, Step 2 is sound, by Lemma A. [External sorting in log space can be done, off-line, by an indexed search of input for "next larger to last output" element, storing only three indices.]

Remark: These times are quadratic due to the $\log(mn)$ space, for one has essentially sorted m n-bit vectors in less than linear space, cf. [7], page 2; without space restrictions we could recognize D_{max} in linear time.

Corollary 2: $D_{\leqq} = \{M \in M_2 : \exists d_\alpha, \ell(d_\alpha) \leq \ell(M), M \models d_\alpha\}$ is decidable in at most quadratic time. [$D_{\leq}$ contains the same matrices as D_{max}.]

Note: By the reduction $M \mapsto (M,\ell(M))$, $D_{\leqq}$ is seen to be essentially easier than D_{par}, which properly contains the "hard" set D^+_{par}, having but one of the latter's restrictions. D_{par} can be solved in $D(LOG^2)$ space, this being the next result. Since D_{par} can clearly be recognized nondeterministically in NP and in (normally) linear space, we claim a significant time/space reduction.

Theorem 3: The "existence of a true, parametric length disjunction" problem-set D_{par} has deterministic time complexity $2^{\log^2(\cdot)}$ [and space complexity $\log^2(\cdot)$].

Proof: Assume w.o.l.o.g. that D_{par} is defined via "$l(d_\alpha) = k$". Suppose $M \in \mathbb{Z}^{m\times n}$ and $k \leq n$ are given; $|C(M)| \leq m$.

0. Compare the (binary) size of numbers mn and 2^k;
   (a) if $mn < 2^k$, answer "Yes":
   comment: since $m < 2^k$, Lemmas A and B apply to any k columns of M, say the first k;

(b) if $mn \geq 2^k$, call the following procedure on (M,k):

1. generate each (ordered) k-tuple S of predicates, one at a time;
2. from S form each possible d_α on these k predicates;
3. evaluate d_α in M and answer "Yes" (& halt) if $M \models d_\alpha$;
4. answer "No".

Steps 1, 2, 3 take times $\binom{n}{k} = O(n^k)$, 2^k, and cmn, respectively. So the total time spent, in case (b) is on the order of

$$mn \cdot 2^k \cdot n^k \leq (mn)^2 \cdot n^{\log(mn)} \leq (mn)^{\log(mn)} = 2^{\log^2(mn)}, \text{ for } m \geq 4.$$

Step 0 takes log(·) space, since we count bits. The generation process in 1, 2 needs $\log^2(\cdot)$ space to store d_α, as k signed indices, each at most n.

Remark 1. This division into cases reduces the wasteful time of the exhaustive procedure (N^N, generally) to a non-exponential worst-case time in (b), where $k \leq \log N$ is crucial, or to a linear time in case (a). The division can be made more simply: (a) $m < 2^k$, and (b) $m \geq 2^k$, are also effective. [Moreover, for half of all parameters, $k \geq \frac{n}{2}$, the non-trivial case (b) holds only for $m \geq 2^{n/2}$.] Using present techniques, a division by comparison of $\binom{n}{k}$ with m, would not give a further (hopeful but fantastic) time reduction of D_{par} into P, because the first case would then fail. Also, while D_{par} is in NP by guessing the d_α, the subsequent evaluation requires the storage of k indices. So nondeterminism seems of no help in lowering the space bound of Theorem 3; e.g., it shouldn't be in NSPACE (log), sometimes called NLOG.

Remark 2. Time-wise, it seems clear that D_{par} is not an NP-complete problem, for in that case we would have NP-Time $\subseteq$ D-TIME($2^{\log^2}$), an unimagineable result; cf [1, problem 10.31.] Also, if it were complete for P under log-space transformations, we would have P $\subseteq$ DSPACE($\log^2$)! Most authors, such as Cook [3], and Jones and Lasser [6], conjecture otherwise. It is still possible D_{par} could be in P, by other methods.

Corollary 3: $D_{\frac{1}{2}}$, the median case of D_{par} (and also, the special case of D_{par} with $k \geq n/2$) can be solved in polynomial time, and log space, deterministically.

Proof: Adapt the proof of Theorem 3, using $k \geq n/2$; since $mn \geq 2^{n/2}$ in case (b), we can bound 2^n thusly: $\binom{n}{k} \leq 2^n \leq (mn)^2$. Store the k-tuples as numbers less than $(mn)^2$, in $2 \cdot \log(mn)$ space. [In face, $D_{\frac{1}{2}}$ takes order N^4 time.]

II. THREE-VALUED CASE

We now investigate the complexity consequences of introducing incompleteness into the data. All the previous matrix problems can be posed for $M \in \mathbb{M}_3$: in this case we denote them with "x". For example, D_{max} becomes $D_{max}(x)$; in most cases, and especially this one, this change makes a problem "harder" (unless it was already log-restricted). Of course if the 2-valued case has intractable complexity, that remains:

$$D^{+}_{par}(x) \text{ and } D^{+}_{\frac{1}{2}}(x) \text{ are NP-complete.}$$

__Theorem 4__: The three-valued case of the problem $D_{max} \in P$, that is $D_{max}(x)$, is NP-complete; furthermore it is so by being P-time isomorphic to the NP-complete problem SAT.

__Proof__: Clearly $D_{max}(x)$ is in NP, by "guessing & verifying". Given a cnf $\mu = \&_{i=1}^{m}(F_i)$, where the F_i are clauses over the set $\{j, \neg j\colon 1 \leqq j \leqq n\}$ of literals, map μ to the matrix $M \in \mathfrak{Z}^{m\times n}$ defined by:

$$M(i,j) = \begin{cases} 1, & \text{if } F_i \text{ contains } j, \\ 0, & \text{if } F_i \text{ contains } \neg j, \\ x, & \text{otherwise.} \end{cases}$$

Since rows correspond to clauses, and columns to atoms, it should be clear that every assignment $\alpha\colon\{1,2,\ldots,n\}\to\{0,1\}$ satisfying μ corresponds uniquely to a maximal length d_α true in M, and conversely. Moreover, the identification map $\mu \to M$ is clearly one-to-one and onto M_3; it inverse is also P-time computable. Both maps can be done in $\log(\cdot)$ space by keeping track only of indices $i,j \leqq n$.

__Corollary 4.1__: $D_{\leqq}(x)$ and $D_{par}(x)$ are NP-complete.

__Corollary 4.2__: The set $D_{\frac{1}{2}}(x)$ is NP-complete.

__Proof__: Reduce $M \in \mathfrak{Z}^{m\times n}$ to $\boxed{M \mid X} \in \mathfrak{Z}^{m\times 2n}$, thereby showing $D_{\leqq}(x) \leqq D_{\frac{1}{2}}(x) \in NP$.

__Corollary 4.3__: The NP-complete 3-SAT problem [1] is P-time (& log space) isomorphic to the "sparse matrix" problem:

$$3\text{-}D_{max}(x) = \{M \in D_{max}(x)\colon M \text{ has at most three 0's or 1's in each row}\}.$$

We now relate SAT to some other problems of intermediate complexity.

__Definition__: We say that a language L is in $P \oplus D(LOG^2)$, or is "P-time-and-$\log^2$-space" decidable, iff there is a single deterministic Turing machine which recognizes L and operates both in polynomial time __and__ $\log^2(\cdot)$ space, as functions of input length. [The generalization to other time-and-space classes is obvious.] The next definition makes the "length" of a cnf somewhat shorter than it is in typical encodings, but this does not affect the results below.

Definition: the length of cnf μ is the number of its literal occurrences and is denoted length (μ).

Lemma 5 (Some Solvable Cases of NP-complete Problems):

A. Any of the following restrictions on parameters ℓ, m, n of an arbitrary cnf $\mu = \&_{i=1}^{m}(D_i)$ on n atoms, ℓ = length (μ), makes SAT = {cnf μ : μ is satisfiable} into a P-time-and-log-space decidable set (SAT_{log}), where u is the number of unit clauses in μ:

(1) $\ell - m \leqq a \log (mn)$ for some $a > 0$;

(2) $m - u \leqq b \log (mn)$ for some $b > 0$;

(3) $\ell - u \leqq c \log (mn)$ for some $c > 0$.

These three conditions are in fact equivalent.

B. For the problem-set 3-SAT = $\{\mu \in SAT : \text{length } (D_i) \leqq 3\}$ any of the conditions (1), (2), (3) implies the weaker condition:

(4) $n_3 \leqq d \cdot \log (mn)$ for some $d \geq 0$,

where n_3 is the number of 3-clauses. The set $3\text{-}SAT_{log}$ = {μ in 3-SAT: (4) holds for μ} is in P and in $D(LOG^2)$ space.

Proof: The equivalence of (1)-(3) is an easy counting argument using $\ell = n_1 + 2n_2 + 3n_3 + \ldots$, $m = n_1 + n_2 + n_3 + \ldots$, and $u = n_1$, where $n_i = |\{i\text{-clauses}\}|$. The proof of part A, with a log(n) space bound is straightforward when (2) is assumed: one generates assignments, $\underline{a}$, for the 2- and 3- clauses, storing indices (only) of the unit-clauses; $|\{\underline{a}\}| \leq (mn)^b$.

In part B, cnf with at most 3-clauses, (2) implies (4) because $m - u = n_2 + n_3$, taking $d = b$.

Suppose that (4) holds for μ, with $d > 0$. [The subcase $d = 0$ gives $SAT_2 = \{\mu \in SAT: \ell(D_i) \leq 2\}$.] Let $x_1, x_2, \ldots, x_k$ represent the distinct atoms occurring in all 3-clauses: $k \leq \log(mn)^{3d}$.

Turing reduction of $3\text{-}SAT_{log}$ to SAT_2:

1. Generate a new assignment $\underline{a}$ to the x_i;
2. Check that each 3-clause has value 1 under $\underline{a}$, and if not return to Step 1;
3. If so, call a procedure to check SAT_2 on the remaining clauses with those literals x_i or $\bar{x}_i$ occurring in 1- and 2-clauses having their values fixed by $\underline{a}$, in order to determine whether these so-modified (via log-space storage) clauses are satisfiable (consistently with $\underline{a}$). If so, then μ is in 3-SAT; if not, return to Step 1;
4. When all possible $\underline{a}$ are exhausted, Halt without accepting.

comment: $UNSAT_2$ is log-complete for $NLOG \subseteq P$, cf. [7].

Hence there exists one deterministic procedure for SAT_2, even with the log-space modifications in Step 3, working in P-time, and another working in $D(LOG^2)$ space.

We have Turing reduced $3\text{-}SAT_{\log}$ to SAT_2, via a log-space, deterministic Turing machine. $3\text{-}SAT_{\log}$ can be placed in P, or in $D(LOG^2)$, because the at most $(mn)^d$ distinct *a* can be exhausted within these bounds.

Remark: The deterministic $UNSAT_2$ procedure in [9] uses linear space; although $UNSAT_2 \in NLOG(\subseteq P)$ can be shown, implying $(UN)SAT_2 \in D(LOG^2)$ by Savitch's Theorem, it is easy to see that the recursion of that theorem has its timing function of order (at least) $n^{\log n}$. We conclude that if either $UNSAT_2$ or SAT_2 is in the deterministic $P \oplus D(LOG^2)$, then so is the harder problem $3\text{-}SAT_{\log}$. But we feel that SAT_2 is not in this simultaneous class, even though it is in $P \cap D(LOG^2)$. Time-space tradeoff is involved. Also, $3\text{-}SAT_{\log}$ is a generalization of SAT_2 which can be (naturally) only Turing reduced to SAT_2.

(MAIN) THEOREM 6:

(a) $SAT_{\log}$ is in DLOG ;

(b) $3\text{-}SAT_{\log}$ is in $P \oplus DLOG^2$ iff SAT_2 is iff $UNSAT_2$ is;

(c) $UNSAT_2 \in P \oplus DLOG^2$ iff $NLOG \subseteq P \oplus DLOG^2$;

(d) $3\text{-}SAT_{\log} \in DLOG$ iff $3\text{-}SAT_{\log}$ is DLOG-complete;

(e) $3\text{-}SAT_{\log} \not\in P \oplus DLOG^2$ implies $P \oplus DLOG^2 \subsetneqq P \cap DLOG^2$.

Proof Outline: (a) = Lemma 5, A; for (b), if $3\text{-}SAT_{\log}$ is in $P \oplus DLOG^2$, then so is its trivial ($n_3 = 0$) case SAT_2, and the converse follows from Lemma 5, B; for (c), use the fact that $UNSAT_2$ is log-complete for NLOG; for (e) observe $(UN)SAT_2 \in P \cap DLOG^2$ and use (b); part (d) says that $3\text{-}SAT_{\log}$ is DLOG-hard, as defined in [7] and is easy to show because $(UN)SAT_2$ is that hard at least.

Remark: One can reduce $3\text{-}UNSAT_{\log} = \{\mu \not\in 3\text{-}SAT\text{:}(4) \text{ holds}\}$ to $UNSAT_2$ via a log-space Turing machine as in Lemma 5; $3\text{-}UNSAT_{\log}$ is thus NLOG-hard. This fact gives a new NLOG-complete problem [with corresponding hypothesis-formation problem for "sparse matrices", for which (4) holds], because : 1) $UNSAT_2$ is in NLOG, and thus so is $3\text{-}UNSAT_{\log}$; 2) $UNSAT_2$ can be trivially reduced to $3\text{-}UNSAT_{\log}$, as its special case with $n_3 = 0$. The complement, $3\text{-}SAT_{\log}$, however, like SAT_2, is in $P \cap D(LOG^2)$ and not, apparently, in NLOG. These problem sets should *help* distinguish NLOG, P, $D(LOG^2)$. We have seen that Turing reductions forced to be time-and-space preserving (iff their called subroutines are) can give partial results, as above.

CONCLUSION: Parts (b), (c) and (e) of Theorem 6 allow one to conclude that either

$$NLOG \subseteq P\oplus DLOG^2 \ [\subseteq P\cap DLOG^2]$$

or else the second inclusion is proper, in which case (easily) P and $DLOG^2$ are incomparable.

Also, one can infer: the problem 3-SAT_{log} is log-complete for $P\oplus D(LOG^2)$ iff the latter class is reducible to NLOG, via log-space reductions, iff $NLOG = P\oplus D(LOG^2)$. Such simulation of NLOG is unlikely. For if 3-(UN)SAT_{log} were log-complete for $P\oplus D(LOG^2)$, then all of these problems, including SAT_2, plus recognition of context-free languages (given a normal-form grammar), could be done in NLOG; cf. [2]. No one seems to believe this is possible either; it would imply that NLOG is closed under complements. The more acceptable alternative conclusion, subject to further study of this evidence, is that NLOG is properly contained in $P\cap DLOG^2$ and is not contained in $P\oplus DLOG^2$. If this is the case, then Cook's conjecture follows.

REFERENCES

[1] Aho, A., Hopcroft J., and Ullman J. : The design and analysis of computer algorithms. Addison-Wesley, 1974.

[2] Cook S.: The complexity of theorem-proving procedures. Proc. 3rd ACM Symposium on Theory of Computing, 1971, 151-158.

[3] Cook S.: An observation on time-storage trade-off, J.Comp. Sys. Sci.9, 308-316 (1974).

[4] Hájek P. Bendoxa K. and Renc Z.: The GUHA method and the three-valued logic, Kybernetika 7(1971), 421-435.

[5] Jones N.D.: Space-bounded reducibility among combinatorial problems, J.Comp.Sys. Sci. 11, 68-85 (1975).

[6] Jones and Laaser W.T.: Complete problems for deterministic polynomial time, Theoretical Computer Science 3 (1977) 105-117.

[7] Jones, Laaser and Lien Y.E.: New problems complete for nondeterministic log space, Mathematical Systems Theory 10 (1976) 1-17.

[8] Karp R.: Reducibility among combinatorial problems, in "Complexity of Computer Computations" (Miller R.E. and Thatcher J.W., eds.), Plenum Press, 1972, 85-103.

[9] Pudlák P.: Polynomially complete problems in the logic of automated research. Mathematical Foundations of Computer Science 1975, Lecture Notes in Computer Science 32, Springer-Verlag, Heidelberg 1975, 358-361.

[10] Pudlák and Springsteel F.N.: Complexity in Mechanized Hypothesis Formation, Theoretical Computer Science, (to appear May, 1979).

SATISFIABILITY PROBLEMS FOR PROPOSITIONAL CALCULI

HARRY R. LEWIS[*]
Harvard University
Cambridge, Massachusetts

ABSTRACT

For each fixed set of Boolean connectives, how hard is it to determine satisfiability for formulas with only those connectives? We show that a condition sufficient for NP-completeness is that the function $x \wedge \sim y$ be representable, and that any set of connectives not capable of representing this function has a polynomial-time satisfiability problem.

With each Boolean function of, say, n variables $a : \{t,f\}^n \rightarrow \{t,f\}$, associate an n-place function sign A. If S is any set of these function signs, or *connectives* as we call them, a term constructed from variables X,Y,...,Z and the connectives in S will be called an S-*formula*, or simply a *formula*. For example, if $S = \{\wedge, \vee, \sim\}$, then the S-formulas are essentially the sentences of the propositional calculus with $\wedge$, $\vee$, $\sim$ as connectives; the only difference is that officially the notation is functional rather then infix. When writing formulas with the standard connectives, however, we use the standard form: $X \wedge Y$ instead of $\wedge(X,Y)$, for example.

Each formula determines, or *represents* as we shall say, a Boolean function of as many arguments as the formula has variables. To fix the correspondence, assume that the variables are ordered, say alphabetically, and let the i-th variable appearing in the formula correspond to the i-th argument of the associated function. In general, we use an uppercase letter such as P to denote a formula, and a lowercase letter such as p for the function represented by that formula. When we wish to enumerate the variables in P explicitly, we may write P as P(X,Y,...,Z), and write P(Q,R,...,S) for the result of substituting other formulas for those variables. Exceptions to the lowercase naming of functions are $\wedge$, $\vee$, $\sim$, etc., which are used to denote functions as well as connectives. Thus, the formula $X \oplus Y$ (or $\oplus(X,Y)$) represents the logical exclusive or function, which is also denoted by $\oplus : \oplus(x,y) = t$ if and only if $x = t$ and $y = f$, or $x = f$ and $y = t$.

If P has n variables X,Y,...,W,...,Z and P(X,Y,...,T,...,Z) and P(X,Y,...,F,...,Z) represent identical functions of $n-1$ arguments,

---

[*] Research supported by NSF Grant No. MCS76-09375-A01.

then W is said to be an *irrelevant* variable of P, otherwise a *relevant* variable. Two formulas that represent identical functions, except perhaps for having different sets of irrelevant variables, will be called *equivalent*. Note that, for example, $X \wedge Y$ and $X \wedge Y \wedge (Z \equiv Z)$ are here considered to be equivalent, even though strictly speaking they do not represent the same function.

A formula $P(X,Y,\dots,Z)$ is said to be *satisfiable* if and only if $p(x,y,\dots,z) = t$ for some $x,y,\dots,z \in \{t,f\}$. The problem addressed here is: how hard is it to determine whether or not an S-formula is satisfiable? Specifically, let S-SAT be the set of all satisfiable S-formulas. By Cook's theorem [1], then, $\{\wedge,\vee,\sim\}$-SAT is NP-complete. In fact, it follows without much difficulty from the work of Cook and Reckhow [2] that for unary and binary connectives S-SAT is NP-complete whenever S is *complete*, i.e. capable of representing all Boolean functions. Our theorem extends and sharpens that result.

MAIN THEOREM. S-SAT *is* NP-*complete provided that there is an* S-*formula equivalent to* $X \wedge \sim Y$. *Otherwise* S-SAT *is decidable in polynomial time*.

This theorem has a family resemblance to Schaefer's result [4]. In our terminology, Schaefer deals with the complexity of determining satisfiability for formulas of the form $A_1(X_1,Y_1,\dots,Z_1) \wedge \cdots \wedge A_n(X_n,Y_n,\dots,Z_n)$, where the X_i, Y_i, and Z_i are variables and $A_1,\dots,A_n$ are connectives drawn from a specified set S.

We begin with the proof of the second assertion. Here we are aided by Post's 1941 analysis of Boolean logic [3]. For any set S of connectives, let $G(S)$ be the set of Boolean functions represented by S-formulas. Post's program was to determine completely the sets $G(S)$ and the relations between them. He showed that if $G(S)$ is not C_1, the set of all Boolean functions, then $G(S)$ is a subset of one of the five sets C_2, C_3, A_1, L_1, and D_3. These sets of functions may be described in terms of sets of connectives. Let

$$C_2^* = \{\wedge,\vee,\supset,\subset,T\}$$

$$C_3^* = \{\wedge,\vee,\not\supset,\not\subset,F\}$$

$$A_1^* = \{\equiv,\oplus,\sim,T,F\}$$

$$L_1^* = \{\wedge,\vee,T,F\}$$

$$D_3^* = \{M_1,M_2\},$$

where

$$m_1(x,y,z) = (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$$

and

$$m_2(x,y,z) = \sim m_1(x,y,z)$$

$$= (\sim x \wedge \sim y) \vee (\sim y \wedge \sim z) \vee (\sim x \wedge \sim z) \quad .$$

Then we may define $C_2 = \mathcal{G}(C_2^*)$, $C_3 = \mathcal{G}(C_3^*)$, $A_1 = \mathcal{G}(A_1^*)$, $L_1 = \mathcal{G}(L_1^*)$, $D_3 = \mathcal{G}(D_3^*)$.

Every C_2^*-formula is satisfiable, in fact if P is such a formula then $p(t,t,\ldots,t) = t$. Therefore if $\mathcal{G}(S) \subseteq C_2$ then satisfiability for S-formulas can be decided trivially.

If X is any relevant variable of an A_1^*-formula $P(\ldots,X,\ldots)$, then $P(\ldots,\sim X,\ldots)$ is equivalent to $\sim P(\ldots,X,\ldots)$. For this reason functions in the class A_1 are called *alternating* by Post; we also call a *formula* alternating if it represents an alternating function. It follows that if $\mathcal{G}(S) \subseteq A_1$ and A is a connective in S, then $A(X,Y,\ldots,Z)$ is alternating. Consequently $A(X,Y,\ldots,Z)$ is equivalent to an extended exclusive or of a constant (T or F) and some of $X,Y,\ldots,Z$. Hence any S-formula can be rewritten as an exclusive or of variables and constants, and by commutativity and associativity this formula can be reduced to one in which no variable appears twice. The original formula was satisfiable unless it reduces in this way to F. This test is clearly polynomial-time.

EXAMPLE. Let $S = \{T,F,\sim,A\}$, where $a(x,y,z,w) = (x \equiv (y \oplus z))$. Consider the S-formula $A(T,\sim A(X,Y,X,Y),X,Y) = P(X,Y)$. Since $(u \equiv v) = (t \oplus u \oplus v)$, and $\sim u = t \oplus u$, $P(X,Y)$ can be transformed into equivalent formulas thus:

$$A(T,\sim A(X,Y,X,Y),X,Y) \ ,$$

$$(T \equiv (\sim A(X,Y,X,Y) \oplus X)) \ ,$$

$$(T \equiv \sim((X \equiv (Y \oplus X)) \oplus X)) \ ,$$

$$T \oplus T \oplus \sim((X \equiv (Y \oplus X)) \oplus X) \ ,$$

$$T \oplus T \oplus T \oplus ((X \equiv Y \oplus X)) \oplus X) \ ,$$

$$T \oplus T \oplus T \oplus ((T \oplus X \oplus (Y \oplus X)) \oplus X) \ ,$$

$$T \oplus T \oplus T \oplus T \oplus X \oplus X \oplus X \oplus Y \ ,$$

$$X \oplus Y \ ,$$

so that P is satisfiable (in fact, $p(t,f) = t$).

Similarly, if $G(S) \subseteq L_1^*$ and A is any connective in S, then $A(X,Y,\ldots,Z)$ is equivalent to a formula involving $T,F,\wedge$, and $\vee$ only. Any S-formula can therefore be reduced as far as possible by repeatedly replacing subformulas representing constants by T or F. The original formula was satisfiable unless this process reduces it to F.

EXAMPLE. Let $S = \{M_1,T,F\}$, where $m_1(x,y,z) = (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$ as defined earlier. Consider the S-formula $M_1(M_1(X,T,F), M_1(F,X,F), M_1(F,F,Y)) = P(X,Y)$. Then $P(X,Y)$ reduces to $M_1(M_1(X,T,F),F,F)$ and then to F, so that $P(X,Y)$ is not satisfiable. (Note that it is important *not* simply to rewrite P using $\wedge,\vee,T,F$, since this could result in an exponential increase in the length of the formula.)

Every D_3^*-formula is satisfiable. This follows from the fact, easily proved by induction on the structure of a formula, that if $P(X,Y,\ldots,Z)$ is a D_3^*-formula and each of $Q,R,\ldots,S$ is either X or $\sim X$, then $P(Q,R,\ldots,S)$ is equivalent to either X or $\sim X$. Therefore, if P is any D_3^*-formula, either $p(t,t,\ldots,t) = t$ or $p(f,f,\ldots,f) = t$. It follows that if $G(S) \subseteq D_3$ then satisfiability is trivially decidable for S-formulas.

The polynomial-time decidability of S-SAT for S such that $G(S)$ is a subset of C_2, A_1, L_1, or D_3 leaves open the question for S such that $G(S) \subseteq C_3$. It turns out that for such S, S-SAT is NP-complete unless its polynomial-time decidability has been established above, i.e., unless $G(S)$ is also a subset of C_2, A_1, L_1, or D_3. We begin by proving in general a sharp form of the result mentioned earlier for unary and binary connectives.

THEOREM 1. *If* S *is complete, then there is a polynomial-time mapping* ϕ_S *from* $\{\wedge,\vee,\sim\}$*-formulas to* S*-formulas such that* P *and* $\phi_S(P)$ *are equivalent for every* $\{\wedge,\vee,\sim\}$*-formula* P. *Consequently, if* S *is complete then* S-SAT *is* NP-*complete*.

Proof. Let S be a complete set of connectives. We begin by showing that S-formulas can be found with certain crucial properties.

LEMMA 1. *There is an* S*-formula* N *equivalent to* $\sim X$ *having only one occurrence of the variable* X.

Proof. Since S is complete there are S-formulas T_S, F_S, and P_S equivalent respectively to T, F, and $\sim X$. Without loss of generality we may assume that T_S and F_S do not contain occurrences of the variable X.

If P_S contains only one occurrence of X then P_S is the required formula N. Otherwise P_S has two or more occurrences of X. Let P_S' be the result of substituting T_S for the first occurrence of X, and let P''_S be the result of substituting F_S for the second occurrence of X. Clearly both P'_S and P''_S have one less occurrence of X than does P_S. We claim that either P'_S or P''_S is equivalent to $\sim X$, which will establish the lemma by induction.

Clearly P'_S is equivalent either to $\sim X$ or to F; for when T_S is substituted for the remaining occurrences of X a formula equivalent to F is obtained. Similarly, P''_S is equivalent either to $\sim X$ or to T. But if P''_S is equivalent to T then P'_S cannot be equivalent to F; for identical formulas can be gotten both from P'_S, by substituting F_S for the first occurrence of X, and from P''_S, by substituting T_S for the first occurrence of X. Hence either P'_S or P''_S is equivalent to $\sim X$. □

LEMMA 2. *There is an* S-*formula* C *equivalent to* $X \wedge Y$ *having only one occurrence of each of the variables* X,Y. *Similarly, there is an* S-*formula* D *equivalent to* $X \vee Y$ *having only one occurrence each of the variables* X,Y.

Proof. A variable X of a formula $P(\ldots,X,\ldots)$ will be called *alternating* if $P(\ldots,\sim X,\ldots)$ is equivalent either to $P(\ldots,X,\ldots)$ or to $\sim P(\ldots,X,\ldots)$. Thus an alternating formula as previously defined is one of which every variable is alternating. Recall also that an alternating *function* is one represented by an alternating formula.

It suffices to show that for some formula $P(X,Y)$ that has two variables X, Y, both nonalternating, there is an S-formula E equivalent to P having only one occurrence each of X and Y. For the two-variable formulas $P(X,Y)$ with nonalternating variables X,Y are exactly those representing functions $p(x,y)$ such that $p(x,y) = t$ for one or three of the pairs (x,y) from among (t,t), (t,f), (f,t), (f,f). If p_1 and p_2 are such functions, then one can be derived from the other by negating one or both arguments, and possibly also the value of the function; for example if $p_1 = \wedge$ and $p_2 = \vee$ then $p_2(x,y) = \sim p_1(\sim x,\sim y)$. Therefore if E can be found as described, then by substitutions the formula N of Lemma 1 can be used to obtain formulas C and D as specified in the statement of the lemma.

No formula can have exactly one nonalternating variable. For clearly any formula with only one variable is alternating. And if $P(X,Y,\ldots,Z,W)$ has alternating variables $X,Y,\ldots,Z$, then a simple induction shows that P is equivalent to $X' \oplus Y' \oplus \cdots \oplus Z' \oplus Q(W)$, where $X',Y',\ldots,Z'$ are the relevant variables among $X,Y,\ldots,Z$, and Q is some formula involving W only. But then W is an alternating variable of Q, and hence of P.

Now since S is complete, S contains some connective A such that $A(X,Y,\dots,Z)$ is not alternating. For a composition of alternating functions is again alternating, and by hypothesis all functions are represented by S-formulas. Thus $A(X,Y,\dots,Z)$ has at least two nonalternating variables.

As in the proof of Lemma 1, let T_S and F_S be S-formulas equivalent to T and F, respectively; without loss of generality we may assume that no variable of T_S or F_S is among $X,Y,\dots,Z$. We now show how to substitute T_S or F_S for all but two of the variables of $A(X,Y,\dots,Z)$ so as to obtain a formula of which the two remaining variables are nonalternating; this will be the required formula E.

First, substitute T_S for each of the alternating variables of $A(X,Y,\dots,Z)$; the result, call it $G(X',Y',\dots,Z')$, is nonalternating in all its relevant variables, which must therefore number at least two. If there are only two, we are done: E may be taken to be G. Otherwise, if G has three or more nonalternating variables, let us rewrite G as $G(X',Y',W',\dots,Z')$, and without loss of generality assume that X',Y', and W' are nonalternating variables of G. We shall substitute T_S or F_S for either X' or Y' in such a way as to obtain a formula that has fewer nonalternating variables than G, but still has at least one (and hence at least two); Lemma 2 will then follow by induction. Let $G_1 = G(T_S,Y',W',\dots,Z')$ and let $G_2 = G(F_S,Y',W',\dots,Z')$. Then G is equivalent to $X \wedge G_1 \vee {\sim}X \wedge G_2$. There are four cases to consider.

(1) Y' is an irrelevant variable of both G_1 and G_2. This case cannot occur, since then Y' would be an irrelevant variable of G.

(2) Y' is a relevant alternating variable of both G_1 and G_2. This case also cannot occur, since then Y' would be an alternating variable of G, as the following series of equivalent formulas demonstrates:

$$G(X',{\sim}Y',W',\dots,Z'),$$

$$X' \wedge G_1({\sim}Y',W',\dots,Z') \vee {\sim}X' \wedge G_2({\sim}Y',W',\dots,Z'),$$

$$X' \wedge {\sim}G_1(Y',W',\dots,Z') \vee {\sim}X' \wedge {\sim}G_2(Y',W',\dots,Z'),$$

$${\sim}G(X',Y',\dots,Z').$$

(3) Y' is a nonalternating variable of G_1 or G_2. Then G_1 or G_2 is the required formula.

(4) Y' is an irrelevant variable of one of G_1, G_2 and a relevant alternating variable of the other. Without loss of generality assume that

Y' is an irrelevant variable of G_1 and a relevant alternating variable of G_2. Then for some formulas H_1 and H_2 not containing X' or Y', G is equivalent to

$$X' \wedge Y' \wedge H_1 \vee X' \wedge \sim Y' \wedge H_1 \vee \sim X' \wedge Y' \wedge H_2 \vee \sim X' \wedge \sim Y' \wedge \sim H_2 \quad .$$

There are two subcases:

(4a) H_1 is equivalent to H_2 or to $\sim H_2$. Then W' cannot be an alternating variable of either H_1 or H_2, otherwise it would be an alternating variable of G. Thus W' is a nonalternating variable of, for example, G_1, which is equivalent to H_1.

(4b) H_1 is equivalent neither to H_2 nor to $\sim H_2$. Then let $G_3 = G(X', T_S, W', \ldots, Z')$; G_3 is equivalent to $X' \wedge H_1 \vee \sim X' \wedge H_2$. Then X' is a nonalternating variable of G_3. □

Now we can complete the proof of Theorem 1. We define a mapping ϕ_S from $\{\wedge, \vee, \sim\}$-formulas to S-formulas thus:

$\phi_S(X) = X$ for any variable X;

$\phi_S(P \wedge Q)$ is the result of substituting $\phi_S(P)$ for X and $\phi_S(Q)$ for Y in C;

$\phi_S(P \vee Q)$ is the result of substituting $\phi_S(P)$ for X and $\phi_S(Q)$ for Y in D;

$\phi_S(\sim P)$ is the result of substituting $\phi_S(P)$ for X in N.

In each case, only a fixed number of new symbols are added, so $\phi_S(P)$ is longer than P by only a fixed amount. Moreover, $\phi_S(P)$ is equivalent to P, and is therefore satisfiable if and only if P is satisfiable. This completes the proof of Theorem 1. □

Now we are ready to show that if there is an S-formula equivalent to $X \not\supset Y$, i.e., to $X \wedge \sim Y$, then S-SAT is NP-complete. First note that $\{\not\supset, T\}$ is complete, since for example, $\not\supset(t,y) = \sim y$, and $\not\supset(x, \not\supset(t,y)) = x \wedge y$. Therefore, $S' = S \cup \{T\}$ is complete. By Theorem 1, then, there is a polynomial-time mapping $\phi_{S'}$ from $\{\wedge, \vee, \sim\}$-formulas to S'-formulas such that P and $\phi_{S'}(P)$ are equivalent for all $\{\wedge, \vee, \sim\}$-formulas P.

Let P be an S-formula equivalent to $X \wedge \sim Y$, and let Q be an arbitrary $\{\wedge, \vee, \sim\}$-formula. Let X_0 be some variable not occurring in

$\phi_{S'}(\sim Q)$, and let R be the S-formula obtained by substituting X_0 for each occurrence of T in $\phi_{S'}(\sim Q)$. By substituting X_0 for X and R for Y in P, we obtain an S-formula Q' equivalent to $X_0 \wedge \sim R$. We claim that Q' is satisfiable if and only if Q is satisfiable, which will complete the proof that S-SAT is NP-complete since Q' can be constructed in time polynomial in the length of Q.

Now Q is satisfiable

if and only if $\sim Q$ is not satisfiable.

if and only if $\phi_{S'}(\sim Q)$ is not satisfiable.

if and only if $\sim\phi_{S'}(\sim Q)$ is satisfiable.

If Q is satisfiable, then so is $\sim\phi_{S'}(\sim Q)$; then so is $X_0 \wedge \sim\phi_{S'}(\sim Q)$, by taking t for the value of X_0; then so is $X_0 \wedge \sim R$, since R is obtained from $\sim\phi_{S'}(\sim Q)$ by substituting X_0 for T; and then so is Q', since Q' is equivalent to $X_0 \wedge \sim R$. Conversely, if Q' is satisfiable, then the value associated with X_0 can only be t, and hence $\sim\phi_{S'}(\sim Q)$ is satisfiable, and hence so is Q. Therefore S-SAT is NP-complete.

It remains only to establish that our results are tight. To do this we again rely on Post's analysis. In his notation, the function $\not\supset$ is in the class F_8^∞ and all the classes of which F_8^∞ is a subset; that is to say if $F_8^\infty \subseteq G(S)$ then $\not\supset$ is a member of $G(S)$, and therefore S-SAT is NP-complete. On the other hand, as can be determined by inspection of the "inclusion diagram" on p. 101 of [3], if $G(S) \subseteq C_3$ but F_8^∞ is not a subset of $G(S)$ then $G(S)$ is a subset of C_2 or A_1, and therefore S-SAT is decidable in polynomial time. This completes the proof of the Main Theorem. □

REFERENCES

[1] S. A. Cook, The complexity of theorem-proving procedures, *Third ACM Symposium on the Theory of Computing*, 1971, pp. 151-158.

[2] S. A. Cook and R. Reckhow, On the length of proofs in the propositional calculus, *Sixth ACM Symposium on the Theory of Computing*, 1974, pp. 135-148.

[3] E. L. Post, The two-valued iterative systems of mathematical logic, *Annals of Mathematics Studies*, Vol. 5, Princeton Univ. Press, 1941.

[4] T. J. Schaefer, The complexity of satisfiability problems, *Tenth ACM Symposium on the Theory of Computing*, 1978, pp. 216-226.

LINEARIZING EFFECT OF FEEDBACK ON NONLINEAR DYNAMICAL SYSTEMS†

C. A. DESOER, Y. T. WANG, AND A. N. PAYNE
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

We study the linearizing effect of feedback on nonlinear, time-varying, multi-input, multi-output, distributed systems. We introduce a nonlinearity measure. With this measure, the condition under which feedback has a linearizing effect is then made precise. An example is provided to illustrate the results.

INTRODUCTION

It is well-known (see e.g.,[2,3,4,5]) that feedback has a linearizing effect on nonlinear systems. Such effect is usually illustrated either by simple examples (as in [4]) or by perturbational analysis (as in [5]). For additional references, see [5]. In this paper, we show that feedback has a linearizing effect on nonlinear systems by first introducing a new concept called the nonlinearity measure. We use the input-output formulation of nonlinear, time-varying, multi-input, multi-output, distributed systems (see e.g.[8]). Such formulation allows for unstable, continuous-time as well as discrete-time subsystems; this is achieved by using causality and the technique of extended spaces, i.e. considering only the time interval [0,T], with T finite but arbitrary.

Notations

Let $\mathbb{R}$ denote the field of real numbers. Let $\mathbb{R}_+$ denote the non-negative real line $[0,\infty)$. Let $\mathcal{T} \subset \mathbb{R}_+$ be the set of time-instants of interest. Let $\underset{\sim}{P}_T$ denote the standard truncation operator. Let $x_T(t) = (\underset{\sim}{P}_T x)(t) = \begin{cases} x, & \text{if } t \leq T \\ 0, & \text{if } t > T. \end{cases}$ Let $|x|_T$ denote $|x_T|$. ":=" means "is defined by".

NONLINEARITY MEASURE

Let $\mathcal{U}_e$ be an extended normed input space. Let $\mathcal{Y}_e$ be an extended semi-normed (output) space. Let $\mathcal{N} = \{\underset{\sim}{N}: \mathcal{U}_e \to \mathcal{Y}_e \mid \underset{\sim}{N}0 = 0,\ \underset{\sim}{N}$ is causal, nonlinear$\}$. Let $\mathcal{L} = \{\underset{\sim}{L}: \mathcal{U}_e \to \mathcal{Y}_e \mid \underset{\sim}{L}$ is causal, linear$\}$. Now consider $\underset{\sim}{N} \in \mathcal{N}$ and $\mathcal{V} \subset \mathcal{U}_e$, a set of inputs of interest. Intuitively, the nonlinearity of $\underset{\sim}{N}$, when $\underset{\sim}{N}$ is driven by $u \in \mathcal{V}$, may be measured by the error $\underset{\sim}{N}u-\underset{\sim}{L}u$ for $u \in \mathcal{V}$, where $\underset{\sim}{L} \in \mathcal{L}$ is a "best linear approximation" of $\underset{\sim}{N}$ over .

Definition (Nonlinearity measure)

Let $\underset{\sim}{N} \in \mathcal{N}, \mathcal{V} \subset \mathcal{U}_e$ and $T \in \mathcal{T}$. Let $|\cdot|$ denote the seminorm associated with $\mathcal{Y}$. The nonlinearity measure of $\underset{\sim}{N}$ over $\mathcal{V}$ with respect to T is the non-negative real number defined by

---

† The complete version of this paper is contained in [1].

Research sponsored by the National Science Foundation Grants ENG76-84522 and ENG73-08214-A01 and the Joint Services Electronics Program Contract F44620-76-C-0100.

$$\delta_T(\underset{\sim}{N},\mathcal{V}) := \inf_{\underset{\sim}{L}\in\mathcal{L}} \sup_{u\in\mathcal{V}} |\underset{\sim}{N}u-\underset{\sim}{L}u|_T \qquad (1)$$

Remarks:

(a) $\underset{\sim}{L}^* \in \mathcal{L}$ is thus said to be a best linear approximation of $\underset{\sim}{N}$ over $\mathcal{V}$ iff $\underset{\sim}{L}^*$ is a minimizer of (1), i.e. $\delta_T(\underset{\sim}{N},\mathcal{V}) = \sup_{u\in\mathcal{V}} |\underset{\sim}{N}u-\underset{\sim}{L}^*u|_T$.

(b) In the case where $\mathcal{Y}_e$ is a seminormed space, we then have the nonlinearity measure of $\underset{\sim}{N}$ over $\mathcal{V}$ with respect to sup $\mathcal{T}$ (typically, sup $\mathcal{T} = \infty$) and eqn. (1) becomes $\delta(\underset{\sim}{N},\mathcal{V}) = \inf_{\underset{\sim}{L}\in\mathcal{L}} \sup_{u\in\mathcal{V}} |\underset{\sim}{N}u-\underset{\sim}{L}u|$.

(c) The well-known describing function (see e.g. [6,7]) is the best linear approximation of a nonlinear operator with respect to our nonlinearity measure (1) provided that $\mathcal{V}$, the set of inputs, is suitably defined. Recall that the criterion which the describing function method uses to find a best linear approximation of $\underset{\sim}{L}$ of a nonlinear system $\underset{\sim}{N}$ is to minimize the mean square error $\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau [(\underset{\sim}{N}u)(t)-(\underset{\sim}{L}u)(t)]^2 dt$ over a class of inputs $u(\cdot)$ (usually $u(t) = a \sin \omega t$, $a > 0$, $\omega > 0$, and thus $\underset{\sim}{L}$ depends on the parameters a, ω). To see the relation between the describing function and our nonlinearity measure, let $a > 0$, $\omega > 0$ be given, let $\mathcal{V}$ be the singleton $\{a \sin \omega t\}$ and $\mathcal{Y}_e = \{y(\cdot):\mathbb{R}_+\to\mathbb{R}^{n_0} \mid y(\cdot)$ is asymptotically $\frac{2\pi}{\omega}$ - periodic$^1\}$ be equipped with the semi-norm $|y| := [\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau |y(t)|^2 dt]^{1/2}$, then a best linear approximation to the nonlinear system $\underset{\sim}{N}$ according to our definition is a minimizer of $\lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau [(\underset{\sim}{N}u)(t)-(\underset{\sim}{L}u)(t)]^2 dt$ which is precisely the describing function of $\underset{\sim}{N}$ with respect to the inputs $u(t) = a \sin \omega t$. Note that in this case the minimizer of (1) (i.e. the describing function of $\underset{\sim}{N}$ with respect to $u(\cdot)$) is parametrized by a and ω.

(d) With the framework of extended spaces, we can discuss the nonlinearity measure of a nonlinear system over a bounded time interval, say, $[0,T]$. Note that a nonlinear system $\underset{\sim}{N}$ may have its nonlinearity measure $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, $\forall T \leq T^* \in \mathcal{T}$, but $\delta_T(\underset{\sim}{N},\mathcal{V}) \neq 0$ for $T > T^*$, simply because $\underset{\sim}{N}$ is operating within the linear range of its characteristics before time T^*.

(e) At the cost of some complication, the class of nonlinear operators under consideration can be extended to include the nonlinear dynamical relations.

Properties of the Nonlinearity Measure $\delta_T(\underset{\sim}{N},\mathcal{V})$

Proposition 1: If $\underset{\sim}{N}_2 = \underset{\sim}{N}_1 + \underset{\sim}{L}_1$ for some $\underset{\sim}{L}_1 \in \mathcal{L}$, then $\delta_T(\underset{\sim}{N}_1,\mathcal{V}) = \delta_T(\underset{\sim}{N}_2,\mathcal{V})$, $\forall T \in \mathcal{T}$, $\forall \mathcal{V} \subset \mathcal{U}_e$.

---

[1] A function $y(\cdot):\mathbb{R}_+\to\mathbb{R}^n$ is said to be asymptotically T-periodic iff $y(\cdot) = y_T(\cdot)+y_o(\cdot)$, where $y_T(\cdot)$ is a T-periodic function and $y_o(t)$ tends to θ_n as $t \to \infty$.

Proposition 2: If $\mathcal{V}_1 \subset \mathcal{V}_2 \subset \mathcal{U}_e$, then $\delta_T(\underset{\sim}{N},\mathcal{V}_1) \le \delta_T(\underset{\sim}{N},\mathcal{V}_2)$, $\forall T \in \mathcal{T}$.

Proposition 3: Suppose that $\forall T \in \mathcal{T}$, $\underset{\sim}{P}_T\mathcal{Y}_e$ is a normed space. U.t.c. if $\underset{\sim}{N}$ is Fréchet differentiable[2] at 0, then $\forall T \in \mathcal{T}$,

$$0 \le \delta_T(\underset{\sim}{N},\bar{B}_T(0;\beta)) \le \sup_{u\in\bar{B}_T(0;\beta)} |\underset{\sim}{N}u - D\underset{\sim}{N}(0)\cdot u|_T \to 0, \text{ as } \beta \to 0 \tag{2}$$

where $\bar{B}_T(0;\beta) := \{u \in \mathcal{U}_e \mid |u|_T \le \beta\}$ and $D\underset{\sim}{N}(0)$ denotes the Fréchet derivative of $\underset{\sim}{N}$ at 0.

Proposition 4: Let $\mathcal{V} \subset \mathcal{U}_e$ be the set of inputs of interest. If for $\underset{\sim}{L} \in \mathcal{L}$, $\underset{\sim}{N}u = \underset{\sim}{L}u$, $\forall u \in \mathcal{V}$, then $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, $\forall T \in \mathcal{T}$. In particular, if $\underset{\sim}{N} \in \mathcal{L}$, then $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, $\forall T \in \mathcal{T}$, $\forall \mathcal{V} \subset \mathcal{U}_e$.

Proposition 5: Let $\mathcal{V} \subset \mathcal{U}_e$ be the set of inputs of interest. Let $\mathcal{L}$ be specialized into the class of continuous[3], linear, causal operators mapping $\mathcal{U}_e$ into $\mathcal{Y}_e$. Suppose that

(a1) $\forall T \in \mathcal{T}$, $\underset{\sim}{P}_T\,\mathcal{Y}_e$ is a Banach space;

(a2) $\forall T \in \mathcal{T}$, $\mathcal{V} \subset \mathcal{U}_e$ is bounded, i.e. $\sup_{u\in} |u|_T < \infty$;

(a3) $\forall T \in \mathcal{T}$, $\exists \beta > 0$ such that $\mathcal{V} \supset \bar{B}_T(0;\beta) := \{u \in \mathcal{U}_e \,\big|\, |u|_T \le \beta\}$.

U.t.c. if for $T \in \mathcal{T}$, $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, then, $\exists \underset{\sim}{L}^* \in \mathcal{L}$ such that

$$|\underset{\sim}{N}u - \underset{\sim}{L}^*u|_T = 0, \ \forall u \in \mathcal{V}. \tag{3}$$

Comments on Propositions 1-5:

(a) Proposition 1 states the obvious fact that if two nonlinear, causal operators differ by a linear causal operator, then they must have the same nonlinearity measure. It is also intuitively clear, from a perturbational viewpoint, that if a linear, causal operator is subject to some nonlinear causal perturbation, then the nonlinearity measure of the perturbed nonlinear, causal operator must be the same as that of the nonlinear perturbation.

(b) Proposition 2 emphasizes the fact that the nonlinearity measure depends on the class of inputs we are considering: the larger the class of inputs we consider, the greater the nonlinearity measure of operator $\underset{\sim}{N}$.

(c) Proposition 3 is another way of stating the well-known fact that (since $\underset{\sim}{N}0 = 0$) the best local linear approximation of a Fréchet differentiable nonlinear operator N at the operating point 0 is the Fréchet derivative of $\underset{\sim}{N}$ at 0. Note that by eqn. (2), $\delta_T(\underset{\sim}{N},B_T(0;\beta)) \to 0$ as $\beta \to 0$, i.e. $\underset{\sim}{N}$ behaves locally, like a linear operator as we expected.

---

[2] $\underset{\sim}{N} \in \mathcal{N}$ is said to be Fréchet differentiable at x iff $\forall T \in \mathcal{T}$, $\underset{\sim}{P}_T\underset{\sim}{N}$ is Fréchet differentiable at x.

[3] $\underset{\sim}{L} \in \mathcal{L}$ is said to be continuous iff $\forall T \in \mathcal{T}$, $\underset{\sim}{P}_T\underset{\sim}{L}$ is continuous, i.e. $\forall T \in \mathcal{T}$,

$$|\underset{\sim}{L}|_T := \sup_{\substack{u\in\mathcal{U}_e\\ |u|_T \neq 0}} \frac{|\underset{\sim}{L}u|_T}{|u|_T} < \infty.$$

(d) Proposition 4 states that $\delta_T(\underset{\sim}{N},\mathcal{V})$ satisfies the natural requirement for a nonlinearity measure, namely, if $\underset{\sim}{N}$ behaves as a linear causal operator over the class of inputs $\mathcal{V}$ in the time interval $[0,T] \subset \mathcal{T}$, then $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$.

(e) With some mild technical assumptions, Proposition 5 establishes the following desirable property of $\delta_T(\underset{\sim}{N},\mathcal{V})$: if $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, then $\underset{\sim}{N}$ behaves like a linear, causal operator over $\mathcal{V}$ in the time interval $[0,T] \subset \mathcal{T}$. Note that if $\delta_T(\underset{\sim}{N},\mathcal{V}) = 0$, then $\delta_{T'}(\underset{\sim}{N},\mathcal{V}) = 0$, $\forall T' \leq T$.

LINEARIZING EFFECT OF FEEDBACK

With the nonlinearity measure defined in eqn. (1), we now can make precise the idea that feedback has a linearizing effect on an otherwise nonlinear system.

Note that the nonlinearity measure defined in (1) allows us to specify quantitatively the extent of nonlinearity of nonlinear system. However, a meaningful comparison requires careful choice of the sets of inputs since the nonlinearity measure depends on the set of inputs we are considering. From an engineering point of view, we are interested in comparing systems which produce desired outputs (e.g. signals of certain dynamical range). Hence, in the following discussion of the linearizing effect of feedback, we shall compare the nonlinearity measure of a nonlinear plant and of a feedback system which consists of such plant; we shall choose a set of inputs for each system so that both systems produce the same set of desired outputs.

Consider the nonlinear, feedback system S shown in Fig. 1, where

$\underset{\sim}{G} : \mathcal{U}_e \to \mathcal{Y}_e$, $\underset{\sim}{K} : \mathcal{R}_e \to \mathcal{U}_e$, $\underset{\sim}{F} : \mathcal{Y}_e \to \mathcal{R}_e$ are nonlinear, causal operators representing the plant, the compensator and the feedback, respectively; $r \in \mathcal{R}_e$ is the system input, $u \in \mathcal{U}_e$ is the plant input, $y \in \mathcal{Y}_e$ is the system output, $e \in \mathcal{R}_e$ is the error signal; $\mathcal{R}_e$, $\mathcal{U}_e$ are extended normed spaces and $\mathcal{Y}_e$ is an extended seminormed space. (4)

We shall assume that

$(\underset{\sim}{I}+\underset{\sim}{F}\underset{\sim}{G}\underset{\sim}{K})^{-1}$ is a well-defined nonlinear, causal operator mapping from $\mathcal{R}_e$ into $\mathcal{R}_e$. (5)

Let $\underset{\sim}{H}_{yr} := \underset{\sim}{G}\underset{\sim}{K}(\underset{\sim}{I}+\underset{\sim}{F}\underset{\sim}{G}\underset{\sim}{K})^{-1}$ denote the closed-loop input-output map.

Let $\mathcal{Y}_{d,e} \subset \mathcal{Y}_e$ be the set of desired outputs.

Let $\mathcal{R}_{d,e} = \{r(\cdot) \in \mathcal{R}_e \mid y = \underset{\sim}{H}_{yr} r \in \mathcal{Y}_{d,e}\} \subset \mathcal{R}_e$. (6)

Let $\mathcal{U}_{d,e} = \{u(\cdot) \in \mathcal{U}_e \mid y = \underset{\sim}{G} u \in \mathcal{Y}_{d,e}\} \subset \mathcal{U}_e$.

Now we have the following theorem.

Theorem 1 (Linearizing effect of feedback)

Consider the nonlinear, feedback system S shown in Fig. 1 and described by eqns. (4)-(6). For $T \in \mathcal{T}$, let $\underset{\sim}{L}^*_G \in \mathcal{L}$ be a best linear approximation to $\underset{\sim}{G}$, i.e.,

$$\delta_T(\underset{\sim}{G},\mathcal{U}_{d,e}) = \sup_{u\in\mathcal{U}_{d,e}} |\underset{\sim}{G}u-\underset{\sim}{L}^*_{G}u|_T \tag{7}$$

Assume that $\underset{\sim}{F}: \mathcal{Y}_e \rightarrow \mathcal{R}_e$ and $\underset{\sim}{K}: \mathcal{R}_e \rightarrow \mathcal{U}_e$ are linear, causal; then

$$\delta_T(\underset{\sim}{H}_{yr},\mathcal{R}_{d,e}) \leq \rho\cdot\delta_T(\underset{\sim}{G},\mathcal{U}_{d,e}) \tag{8}$$

where

$$\rho =: \sup_{\substack{y\in\tilde{\mathcal{Y}}_e \\ y_T\neq 0}} \frac{|(\underset{\sim}{I}+\underset{\sim}{L}^*_G\underset{\sim}{K}\underset{\sim}{F})^{-1}y|_T}{|y|_T} \tag{9}$$

with $\tilde{\mathcal{Y}}_e := (\underset{\sim}{G}-\underset{\sim}{L}^*_G)\mathcal{U}_{d,e}$.

Remarks:

(a) In a design problem, given some $\underset{\sim}{G} \in \mathcal{N}$ together with its best linear approximation $\underset{\sim}{L}^*_G$ over $\mathcal{U}_{d,e}$ with respect to some $T \in \mathcal{T}$, if one designs $\underset{\sim}{K}$, $\underset{\sim}{F}$ such that ρ be much less than 1, consistent with other requirements, then by inequality (8), $\delta_T(\underset{\sim}{H}_{yr},\mathcal{R}_{d,e}) \ll \delta_T(\underset{\sim}{G},\mathcal{U}_{d,e})$; i.e. for the class of inputs under consideration and for the time interval of interest, the closed-loop system is much closer to a linear system than $\underset{\sim}{G}$ itself. This result clearly exhibits the linearizing effect of feedback.

(b) Note that ρ is defined via the (linearized) inverse return difference operator $[\underset{\sim}{I}+\underset{\sim}{L}^*_G\underset{\sim}{K}\cdot\underset{\sim}{F}]^{-1}$ (when we break the loop after the plant $\underset{\sim}{G}$) as we expected [5]: since the nonlinear plant $\underset{\sim}{G}$ can be thought as a linear plant $\underset{\sim}{L}^*_G$ being subjected to some nonlinear perturbation $\underset{\sim}{G}-\underset{\sim}{L}^*_G$ and the effect of an (infinitesimal) nonlinear perturbation on the otherwise linear closed-loop system is reduced by the factor $(\underset{\sim}{I}+\underset{\sim}{L}^*_G\underset{\sim}{K}\underset{\sim}{F})^{-1}$ by feedback.

(c) If $\underset{\sim}{L}^*_G$, $\underset{\sim}{K}$, $\underset{\sim}{F}$ are linear, time-invariant and represented by transfer function matrices $L^*_G(j\omega)$, $K(j\omega)$, $F(j\omega)$, respectively, then $\rho \ll 1$ if the maximum singular value of $[1+L^*_G(j\omega)K(j\omega)F(j\omega)]^{-1}$ is much less than 1 over the frequencies of interest.

EXAMPLE

Consider the nonlinear, feedback system S shown in Fig. 1, where $\underset{\sim}{G}$ is characterized by a piecewise-linear function shown in Fig. 2, $\underset{\sim}{K}$ and $\underset{\sim}{F}$ are represented by constant gains 10 and 1, respectively. It is easy to show that the closed-loop input-output map $\underset{\sim}{H}_{yr}$ is characterized by the piecewise-linear function shown in Fig. 3. Now let us consider the case where $\mathcal{Y}_{d,e} = \{y(\cdot):\mathbb{R}_+\rightarrow\mathbb{R} \,\big|\, |y|_\infty \leq 0.8\}$, then the corresponding $\mathcal{U}_{d,e} = \{u(\cdot):\mathbb{R}_+\rightarrow\mathbb{R} \,\big|\, |u|_\infty \leq 1.2\}$ and $\mathcal{R}_{d,e} = \{r:\mathbb{R}_+\rightarrow\mathbb{R} \,\big|\, |r|_\infty \leq 0.92\}$. A straightforward minmax calculation shows that the best linear approximation $\underset{\sim}{L}^*_G$ of $\underset{\sim}{G}$ is a constant gain of 0.6 and the nonlinearity measure of $\underset{\sim}{G}$ is

$\delta_T(\underset{\sim}{G},\mathcal{U}_{d,e}) = 0.12$, $\forall T \in \mathcal{T}$. Similarly, $\delta_T(\underset{\sim}{H}_{yr},\mathcal{R}_{d,e})$

$= \sup_{r\in\mathcal{R}_{d,e}} |\underset{\sim}{H}_{yr}r - \frac{6}{7}r|_{\infty,T} = \frac{0.12}{7}$, $\forall T \in \mathcal{T}$. Thus the nonlinearity measure of $\underset{\sim}{G}$ has been reduced by 7 by feedback. Note that $\rho = \frac{1}{1+0.6\text{x}10} = \frac{1}{7}$, i.e. for this example, the equality holds in (8). The best linear approximations of $\underset{\sim}{G}$ and $\underset{\sim}{H}_{yr}$ are shown, by the broken lines, in Fig. 2 and Fig. 3 respectively. To further illustrate the linearizing effect of feedback, we drive the nonlinear plant $\underset{\sim}{G}$ with $u(t) = 1.2 \sin \omega t$ and the closed-loop system $\underset{\sim}{G}\underset{\sim}{K}(\underset{\sim}{I}+\underset{\sim}{F}\underset{\sim}{G}\underset{\sim}{K})^{-1}$ with $r(t) = 0.92 \sin \omega t$. The corresponding (open-loop system) $\underset{\sim}{y}_0$ and the (closed-loop) output y are shown in Fig. 4.

CONCLUSION

We defined a new concept called the nonlinearity measure and then obtained condition under which feedback has a linearizing effect. Theorem 1 showed that the nonlinearity measure of a feedback system is bounded above by the product of the nonlinearity measure of the nonlinear plant and the gain of the (linearized) inverse return difference operator. Hence, for the very general class of systems considered in this paper, appropriately designed feedback can make the closed-loop system much more linear than the nonlinear plant.

REFERENCES

[1] C. A. Desoer and Y. T. Wang, "Foundations of feedback theory for nonlinear dynamical systems," submitted to IEEE Trans. on Circuits and Systems and the 1979 International Symposium on Circuits and Systems.

[2] H. S. Black, "Stabilized Feedback Amplifiers," The Bell System Technical Journal, pp. 1-18, Jan. 1934.

[3] I. M. Horowitz, Synthesis of Feedback Systems, Academic Press, New York, 1963.

[4] J. K. Roberge, Operational Amplifiers: Theory and Practice, John Wiley and Sons, Inc., New York, 1975.

[5] J. B. Cruz, Jr., "Effect of feedback on signal distortion in nonlinear systems," in Feedback Systems (ed. by J. B. Cruz, Jr.), New York: McGraw-Hill, 1972, Chap. 3.

[6] A. Gelb and W. E. Vander Velde, Multi-input Describing Functions and Nonlinear System Design, McGraw-Hill, Inc., New York, 1968.

[7] M. Vidyasagar, Nonlinear System Analysis, Prentice-Hall, New Jersey, 1978.

[8] C. A. Desoer and M. Vidyasagar, Feedback Systems: Input-Output Properties, Academic Press, New York, 1975.

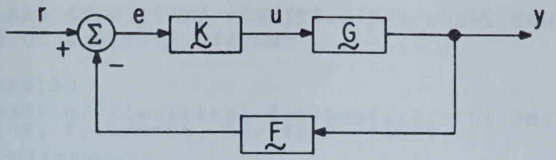


Fig. 1. Nonlinear, feedback system S under consideration.

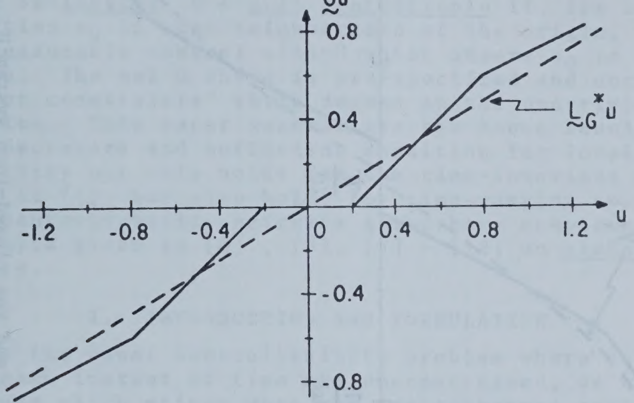


Fig. 2. Characterizations of the nonlinear plant $\underset{\sim}{G}$ and its best linear approximation $\underset{\sim}{L}_G^*$ (in broken lines).

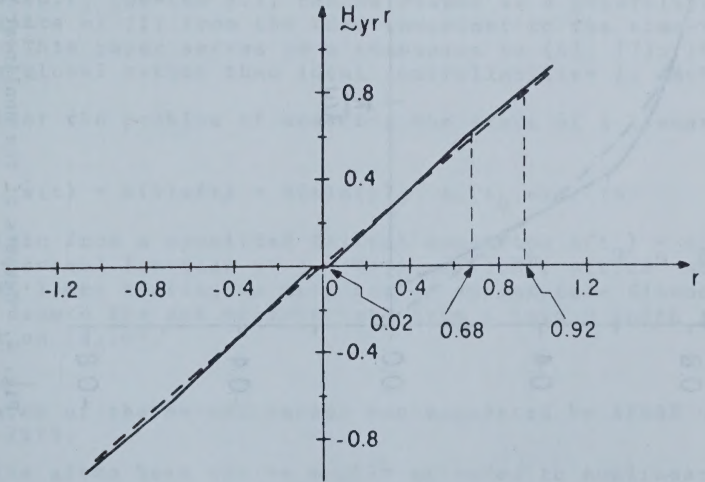


Fig. 3. Characterizations of the closed-loop system $\underset{\sim}{H}_{yr}$ and its best linear approximation (in broken lines).

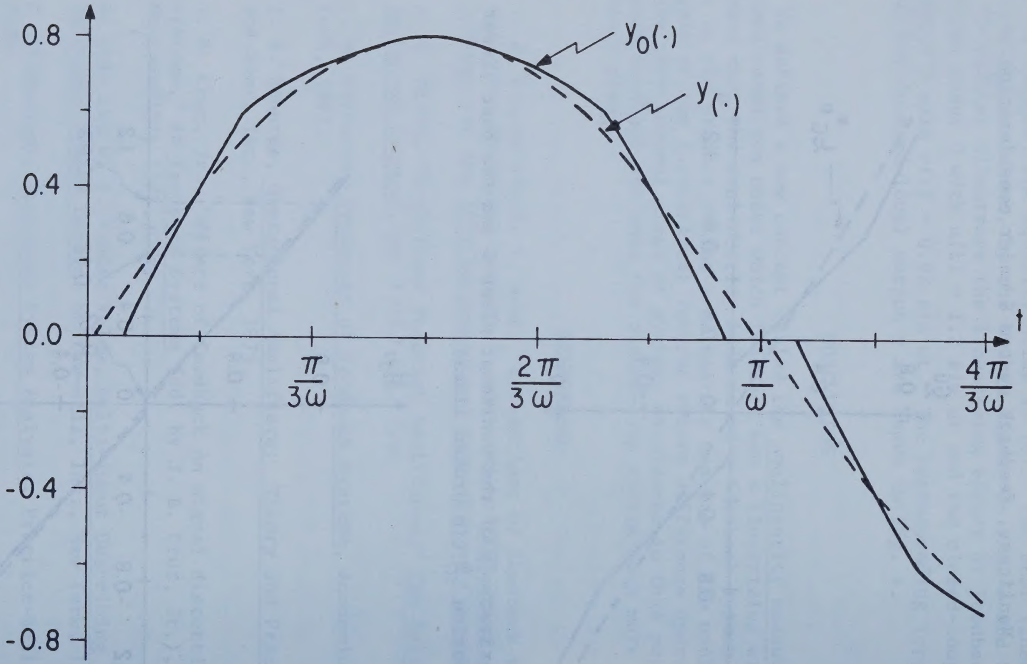


Fig. 4. Outputs of the nonlinear plant $\tilde{G}$, y_0, and the closed-loop system $\tilde{H}_{yr}$, y, when the plant input $u(t) = 1.2 \sin \omega t$ and the closed-loop system input $r(t) = 0.92 \sin \omega t$.

A NECESSARY AND SUFFICIENT CONDITION FOR LOCAL CONSTRAINED CONTROLLABILITY OF A LINEAR SYSTEM[1]

B.R. BARMISH
Department of Electrical Engineering, The University of Rochester, Rochester, New York 14627

W.E. SCHMITENDORF
Department of Mechanical Engineering, Northwestern University, Evanston, Illinois 60201

Abstract. The linear state equation $\dot{x}(t) = A(t)x(t) + B(t)u(t)$ is said to be locally Ω - null controllable if, for every initial condition x_0 in some neighborhood of the origin, there exists a measurable control $u(t)\varepsilon\Omega$ which steers x_0 to zero in finite time. The set Ω above is pre-specified and corresponds to "actuator constraints" which depend on the underlying physical problem. This paper generalizes the known result of [1]. i.e., our necessary and sufficient condition for local Ω - null controllability not only holds for the time-invariant systems considered in [1], but also holds for time-varying systems. The local controllability criteria given here complement a number of results given in [6] , [7], [9] - [12] on global controllability.

I. INTRODUCTION AND FORMULATION

Unlike the usual controllability problem where control values at each instant of time are unconstrained, we consider here the case which arises when the instantaneous control values belong to a prescribed set Ω in R^m. Constrained controllability problems, for linear time-invariant systems, are examined in [1] - [5]. In this paper, our definition of local constrained controllability is identical to that of Brammer [1]. Our main result, Theorem 2.1, can be viewed as a generalization of the results of [1] from the time-invariant to the time-varying case. This paper serves as a companion to [6], [7], [9] - [12] where global rather than local controllability is emphasized.

Consider the problem of steering the state of a linear system[†]

$$\dot{x}(t) = A(t)x(t) + B(t)u(t); \quad t\varepsilon[t_0,\infty) \qquad (S)$$

to the origin from a specified initial condition $x(t_0) = x_0$ by choice of control function $u(\cdot)$. Here, $x(t)\varepsilon R^n$, $u(t)\varepsilon R^m$ and $A(\cdot)$ and $B(\cdot)$ are continuous matrices of appropriate dimension. Let $M(\Omega)$ denote the set of functions from R into Ω which are measurable on $[t_0,\infty)$.

---

[1] The research of the second author was supported by AFOSR under Grant 76-2923.

[†] The results given here can be easily extended to nonlinear systems of the form $\dot{x}(t) = A(t)x(t) + f(t,u(t))$ -- provided that f is measurable in its first argument and continuous in its second argument.

Definitions: (S) is said to be *constrained controllable* (more precisely, Ω - *null controllable*) *at* (x_0,t_0) if given any initial condition $x(t_0) = x_0$, there exists an input $u(\cdot)\varepsilon M(\Omega)$ such that the solution $x(\cdot)$ of (S) satisfies $x(T) = 0$ for some $T\varepsilon[t_0,\infty)$. We say that (S) is *locally* Ω - *null controllable* (*at* t_0) if there exists an open set $V\subset R^n$ containing the origin, such that (S) is Ω - null controllable at (x_0,t_0) for all $x_0\varepsilon V$. If V can be taken to be all of R^n, then (S) is said to be *globally* Ω - *null controllable* at t_0.

Remarks: We note that a system may be completely controllable in the usual sense (globally R^m - null controllable in our notation) but *not* locally Ω - null controllable. This fact is illustrated by the simple scalar example

$$\dot{x}(t) = x(t) + u(t); \quad t\varepsilon[0,\infty); \quad \Omega = [0,1].$$

(If the initial condition x_0 is positive, then it cannot be steered to zero if the input is constrained by $\Omega = [0,1]$.)

Result of Brammer: Consider the case when $A(t) \equiv A$ and $B(t) \equiv B$ are time-invariant. Furthermore, assume Ω satisfies the following conditions: there exists a $u\varepsilon\Omega$ such that $Bu = 0$ and the convex hull of Ω has non-empty interior in R^m. For this special case, our main result (to be described in the sequel) is comparable to the following result given in [1].

Theorem 1.1: (S) *is locally* Ω - *null controllable at* t_0 *if and only if*

(i) rank $[B:AB:A^2B:\ldots A^{n-2}B:A^{n-1}B] = n$;

(ii) there is *no real eigenvector* v *of* A' *satisfying* $v'B\omega\leq 0$ *for all* $\omega\varepsilon\Omega$.

II. MAIN RESULT

Notational Preliminaries: Our necessary and sufficient condition for local Ω - null controllability will be expressed in terms of $A(\cdot)$, $B(\cdot)$ and $H_\Omega: R^m\to R$, the support function on Ω, which is given by

$$H_\Omega(\alpha) \triangleq \sup\{\omega'\alpha:\omega\varepsilon\Omega\}$$

for $\alpha\varepsilon R^m$. We need some further notation: Suppose $z(\cdot)$ is a nontrivial solution of the adjoint system

$$\dot{z}(t) = -A'(t)z(t); \quad t\varepsilon[t_0,\infty). \qquad \text{(S')}$$

Then the *normalized solution* $z_*(\cdot)$ *of* $z(\cdot)$ is given by

$$z_*(t)\triangleq\frac{z(t)}{||z(0)||}.$$

Finally, $\phi(t,\tau)$ will denote the state transition matrix for (S).

Theorem 2.1: *Suppose* Ω *is compact and contains zero. Then* (S) *is locally* Ω - *null controllable at* t_0 *if and only if there exists an* $\varepsilon>0$ *such that*

$$\int_{t_0}^{\infty} H_\Omega(B'(\tau)z_*(\tau))d\tau \geq \varepsilon \qquad (2.1)$$

for all normalized non-trivial solutions $z_*(\cdot)$ of (S'). *Equivalently, if and only if*

$$\int_{t_0}^{\infty} \sup\{\omega' B'(\tau)\phi'(t_0,\tau)\lambda : \omega\varepsilon\Omega\} d\tau \geq \varepsilon \qquad (2.1a)$$

for all $\lambda\varepsilon R^n$, $||\lambda|| = 1$.

In order to prove the theorem, we shall make use of two fundamental lemmas whose proofs are given in [6] and [7].

Lemma 2.2: *The integral operations in* (2.1) *and* (2.2) *are well defined.* i.e., $H_\Omega(B'(\tau)z_*(\tau))$ *is a non-negative measurable function* (hence integrable) *along all normalized trajectories* $z_*(\cdot)$ *of* (S').

Lemma 2.3: *Let* Ω *satisfy the hypothesis of Theorem* 2.1. *Then* (S) *is* Ω - *null controllable at initial epoch* (x_0,t_0) *if and only if there exists some* $T\varepsilon[t_0,\infty)$ *such that*

$$x_0'z(t_0) + \int_{t_0}^{T} H_\Omega(B'(\tau)z(\tau))d\tau \geq 0 \qquad (2.3)$$

for all solutions $z(\cdot)$ *of* (S').

Proof of Theorem 2.1: Without loss of generality, we henceforth take $t_0 = 0$ for notational convenience.

(Sufficiency): Suppose that the integral condition in (2.1) is satisfied for some $\varepsilon > 0$. Let

$$V \triangleq \{x\varepsilon R^n : ||x|| < \tfrac{\varepsilon}{4}\}.$$

Then V is an open neighborhood which contains zero; and we claim that (S) is Ω - null controllable at (x_0,t_0) for all $x_0\varepsilon V$. To establish this claim, we pick any $x_0\varepsilon V$ and it suffices to show, by Lemma 2.3, that there exists some $T\varepsilon[0,\infty)$ such that

$$x_0'z(0) + \int_0^T H_\Omega(B'(\tau)z(\tau))d\tau \geq 0 \qquad (2.4)$$

for all solutions $z(\cdot)$ of (S'). We proceed by contradiction. i.e., we assume that no such $T\varepsilon[0,\infty)$ exists. Hence, there exists a monotone sequence $T_n \to \infty$ and a corresponding sequence $\langle z^n(0)\rangle_{n=1}^{\infty}$ of non-zero starting conditions for (S') such that

$$x_0'z^n(0) + \int_0^{T_n} H_\Omega(B'(\tau)z^n(\tau))d\tau < 0;\ n = 1, 2, \ldots$$

where $z^n(\cdot)$ above is the solution of (S') resulting from the initial condition $z(0) \triangleq z^n(0)$. Now, the normalized sequence

$<z_*^n(0)>_{n=1}^{\infty}$ remains within the compact set

$$\Sigma \triangleq \{x \varepsilon R^n: ||x|| = 1\} .$$

Consequently, by the compactness of Σ, we can extract a convergent subsequence $z_*^{n_j}(0)$ which converges to some vector $\tilde{z}(0) \varepsilon \Sigma$. i.e., $||\tilde{z}_*(0)|| = 1$. We notice that the corresponding subsequence of times, $<T_{n_j}>_{n_j=1}^{\infty}$, still tends to $+\infty$. For each n_j, we have, by (2.4)

$$\int_0^{T_{n_j}} H_\Omega(B'(\tau)z_*^{n_j}(\tau))d\tau < -x_0'z_*^{n_j}(0); \ n_j = 1, 2, \ldots$$

Since $0 \varepsilon \Omega$, the lefthand side is non-negative; and it follows that

$$\int_0^{T_{n_j}} H_\Omega(B'(\tau)z_*^{n_j}(\tau))d\tau < ||x_0||\cdot||z_*^n(0)||$$

$$= ||x_0||$$

$$< \frac{\varepsilon}{4} ; \ n_j = 1, 2, \ldots . \qquad (2.5)$$

We would like to obtain an inequality involving $\tilde{z}_*(\cdot)$ with an infinite upper limit on the integral in (2.5) above. To accomplish this, we define

$$f_{n_j}(\tau) \triangleq H_\Omega(B'(\tau)z_*^{n_j}(\tau))d\tau \quad \text{if} \quad \tau\varepsilon[0,T_{n_j}];$$

$$\triangleq 0 \text{ otherwise.}$$

$$f(\tau) \triangleq H_\Omega(B'(\tau)\tilde{z}_*(\tau))$$

and make the following observations:

(i) By (2.5), $\int_0^\infty f_{n_j}(\tau)d\tau$ is bounded by $\frac{\varepsilon}{4}$ for $n_j = 1,2,\ldots$

(ii) $f_{n_j}(\tau)$ converges pointwise to $f(\tau)$ on $[0,\infty)$.

This observation is proven using the facts that

$z_*^{n_j}(0) \to \tilde{z}_*(0)$; $T_{n_j} \to \infty$; the solution of (S') depends continuously on the initial condition; $H_\Omega(\cdot)$ depends continuously on its argument.

Applying Fatou's Lemma [8] to the $f_{n_j}(\cdot)$, we have

$$\int_0^\infty f(\tau)d\tau \leq \liminf_{nj \to \infty} \int_0^\infty f_{n_j}(\tau)d\tau$$

$$\leq \limsup_{n_j \to \infty} \int_0^\infty f_{n_j}(\tau)d\tau$$

$$< \frac{\varepsilon}{4} \quad .$$

Substitution for $f(\tau)$ above gives

$$\int_0^\infty H_\Omega(B'(\tau)\tilde{z}_*(\tau))d\tau < \frac{\varepsilon}{4} \quad .$$

Hence $\tilde{z}_*(\cdot)$ is a normalized solution of (S') which violates the integral condition in (2.1). This is the contradiction which we seek.

(Necessity): We assume that (S) is locally Ω - null controllable at t_0; we must verify the integral condition in (2.1). By hypothesis, there exists an open neighborhood

$$V_{2\varepsilon} \triangleq \{x\varepsilon R^n\colon \; ||x|| < 2\varepsilon\}; \varepsilon > 0$$

such that (S) is Ω - null controllable at (x_0,t_0) for every $x_0 \varepsilon V_{2\varepsilon}$. Hence, (S) is Ω - null controllable at (x_0,t_0) for all x_0 in the set

$$\overline{V}_\varepsilon \triangleq \{x\varepsilon R^n\colon \; ||x|| \leq \varepsilon\}.$$

For this choice of $\varepsilon > 0$, we claim that the integral condition in (2.1) will hold for all normalized solutions $z_*(\cdot)$ of (S'). To prove this claim, let $z(\cdot)$ be a non-zero trajectory of (S'). We have

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = \frac{1}{||z(0)||}\int_0^\infty H_\Omega(B'(\tau)z(\tau))d\tau$$

$$= \frac{1}{||z(0)||}\left[x_0'z(0) + \int_0^\infty H_\Omega(B'(\tau)z(\tau))d\tau\right] - \frac{x_0'z(0)}{||z(0)||} \quad (2.6)$$

for all $x_0 \varepsilon \overline{V}_\varepsilon$. Since, we are free to pick x_0 in the equality above, we make the following special choice: We pick $\tilde{x}_0 \varepsilon \overline{V}_\varepsilon$ such that

$$\frac{\tilde{x}_0' z(0)}{||z(0)||} = -\varepsilon .$$

(We note that such a selection is always possible because $z(0)/||z(0)||$ is a unit vector and V_ε is a ball of radius $\varepsilon > 0$.) For this choice of $\tilde{x}_0$, by Lemma 2.3, there exists some $\tilde{T}\varepsilon[0,\infty)$ such that

$$\tilde{x}_0' z(0) + \int_0^{\tilde{T}} H_\Omega(B'(\tau)z(\tau))d\tau \geq 0. \qquad (2.7)$$

Returning to (2.6), we now have

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = \frac{1}{||z(0)||}[\tilde{x}_0' z(0) + \int_0^\infty H_\Omega(B'(\tau)z(\tau))d\tau]$$

$$- \frac{\tilde{x}_0' z(0)}{||z(0)||}$$

$$= \frac{1}{||z(0)||}[\tilde{x}_0' z(0) + \int_0^\infty H_\Omega(B'(\tau)z(\tau))d\tau] + \varepsilon$$

$$\geq \frac{1}{||z(0)||}[\tilde{x}_0' z(0) + \int_0^{\tilde{T}} H_\Omega(B'(\tau)z(\tau))d\tau] + \varepsilon$$

(using $H_\Omega \geq 0$)

$$\geq \varepsilon \quad \text{(using (2.7))}.$$

The equivalent condition (2.1a) is obtained from viewing λ as an initial condition for a normalized trajectory of (S'). This completes the proof of the theorem. □

III. EXAMPLES

To illustrate the usage of Theorem 2.1, we consider the time-varying system (S) described by

$$\dot{x}_1(t) = -tx_1(t) - e^{-\frac{t^2}{2}} x_2(t) ;$$

$$\dot{x}_2(t) = -x_2(t) + u(t) ; \; t\varepsilon[0,\infty).$$

To decide whether or not (S) is locally Ω - null controllable, we form the adjoint system

$$\dot{z}_1(t) = t z_1(t) \quad ;$$

$$\dot{z}_2(t) = e^{\frac{-t^2}{2}} z_1(t) + z_2(t) \quad ; \; t\varepsilon[0,\infty) \quad (S')$$

whose solution is given by

$$z_1(t) = e^{\frac{t^2}{2}} z_1(0) \quad ;$$

$$z_2(t) = (e^t-1)z_1(0) + e^t z_2(0).$$

Consider $\Omega = [-M,M]$ where $M > 0$. Then, in accordance with the theorem, we must look at

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = \int_0^\infty \sup_{|\omega| \le M} \left([0 \quad 1] \begin{bmatrix} e^{\tau^2/2} & 0 \\ e^\tau - 1 & e^\tau \end{bmatrix} z_*(0)\right)d\tau$$

$$= \int_0^\infty |(e^\tau - 1)z_{1*}(0) + e^\tau z_{2*}(0)| \, d\tau$$

where $z_*(0) \triangleq [z_{1*}(0)z_{2*}(0)]'$ has unit norm. Hence,

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = M \int_0^\infty |e^t(z_{2*}(0)) - z_{1*}(0)) - z_{1*}(0)| \, d\tau$$

Now, there are two cases to consider:

Case 1: $\eta \neq 0$ where $\eta \triangleq z_{2*}(0) - z_{1*}(0)$.

In this case, the integrand tends to $+\infty$ and any lower bound $\varepsilon > 0$ will be valid for the integral of $H_\Omega(B'(\tau)z_*(\tau))$.

Case 2: $\eta = 0$. In this case, we have

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = M \int_0^\infty |z_{1*}(0)|d\tau.$$

We note that $z_{1*}(0)$ cannot be zero; otherwise $z_{2*}(0)$ would also be zero; and this would contradict the fact that $||z_*(0)|| = 1$. Hence, we conclude that

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = +\infty$$

and once again any lower bound $\varepsilon > 0$ will be valid for the the integral of $H_\Omega(B'(\tau)z_*(\tau))$.

From the theorem, it follows that (S) is indeed locally Ω-null controllable. By Theorem 2.1 of [6], it can also be shown that (S) is also globally Ω - null controllable.

An example of a system which is locally but not globally Ω-null controllable is given by

$$\dot{x}(t) = tx(t) + u(t); \; \Omega = [-1,1].$$

Then, for each initial condition $z(0)$ of (S'), we obtain a normalized trajectory $z_*(t)$ which is given by

$$z_*(t) = \frac{e^{-\frac{t^2}{2}} z(0)}{|z(0)|} .$$

For this example, (2.1) becomes

$$\int_0^\infty H_\Omega(B'(\tau)z_*(\tau))d\tau = \int_0^\infty \sup_{|\omega|\leq 1} \frac{\omega e^{-\frac{\tau^2}{2}} z(0)}{|z(0)|} \, d\tau$$

$$= \int_0^\infty e^{-\frac{\tau^2}{2}} d\tau$$

$$= \sqrt{\frac{\pi}{2}}$$

Consequently, condition (2.1) is satisfied by taking

$$0 < \varepsilon \leq \sqrt{\frac{\pi}{2}} .$$

IV. CONCLUDING REMARKS

The Set Ω : In contrast to much of the existing literature, Theorem 2.1 does not require zero to be interior to Ω or its convex hull. Our weaker assumption "$0\varepsilon\Omega$" plays an important technical role: It ensures non-negativity (hence integrability) of $H_\Omega(B'(\tau)z_*(\tau))$ in (2.1). This assumption is not required, however, when developing the theory for Ω - null controllability at a specific initial epoch (x_0,t_0). Results to this effect are given in [6] and [7]. We also note that although Ω was assumed compact, Theorem 2.1 can still be proven with $\Omega = R^m$. For this "unconstrained case", the theorem naturally degenerates into the known results for unconstrained controllability, i.e., (S) is locally R^m - null controllable if and only if the controllability matrix has full rank.

Equivalent Control Sets: We know that if Ω' is another compact set having the same convex closure as Ω, then

$$H_\Omega(\alpha) = H_{\Omega'}(\alpha) \qquad \underline{\text{for all}}\ \alpha \varepsilon R^m .$$

As a consequence of this fact in conjunction with Theorem 2.1, we conclude that (S) *is locally* Ω - *null controllable if and only if* (S) *is locally* Ω' - *null controllable*. Consequently, when carrying out computations, we are free to replace an "irregular" Ω by a "regular" Ω'.

Additional Structure on Ω: Some interesting special cases of Theorem 2.1 arise when Ω is endowed with additional structure (over and above "$0\varepsilon\Omega$" and "Ω compact"). For example, if we take

$$\Omega = \{\omega \varepsilon R^m : ||\omega|| \leq 1\} ,$$

then integral condition (2.1) becomes

$$\begin{aligned} \varepsilon &\leq \int_{t_0}^{\infty} \sup \{\omega' B'(\tau) z_*(\tau) : ||\omega|| \leq 1\}\, d\tau \\ &= \int_{t_0}^{\infty} ||\, B'(\tau) z_*(\tau) \,||\, d\tau . \end{aligned} \tag{4.1}$$

This special case now looks quite similar to the criteria in [4] and [5] for global Ω - null controllability.

Finite-Dimensionality: Theorem 2.1 demonstrates that the question of local Ω - null controllability can be answered by solving a finite-dimensional problem in the variable $z_*(0)\varepsilon R^n$ or equivalently in the variable $\lambda\varepsilon R^n$. From the point of view of numerical computation, a search over $||\lambda|| = 1$ is certainly preferrable to a search over the infinite-dimensional set $M(\Omega)$.

VI. REFERENCES

[1] R. Brammer, "Controllability of Linear Autonomous Systems with Positive Controllers," *SIAM J. Control*, vol. 10, pp. 339-353, 1972.

[2] S. Saperstone and J. Yorke, "Controllability of Linear Oscillatory Systems Using Positive Controls," *SIAM J. Control*, vol. 9, pp. 253-262, 1971.

[3] S. Saperstone, "Global Controllability of Linear Systems with Positive Controls," *SIAM J. Control*, vol. 11, pp. 417-423, 1973.

[4] R. Conti, *Teoria del Controllo e del Controllo Ottimo*, UTET, Torino, Italy, 1974.

[5] L. Pandolfi, "Linear Control Systems: Controllability with Constrained Controls," *J. Optimization Theory and Appl.*,

vol. 19, pp. 577-585, 1976.

[6] W.E. Schmitendorf and B.R. Barmish, "Constrained Controllability of Linear Systems," to appear in print.

[7] B.R. Barmish and W.E. Schmitendorf, "Constrained Controllability," to appear in the Proceedings of the IEEE Conference on Decision and Control, San Diego, 1978.

[8] H.L. Roydon, Real Analysis, Macmillan Co., New York, New York, 1968.

[9] J.P. Dauer, "Controllability of Nonlinear Systems Using a Growth Condition," J. Optimization Theory and Appl., vol. 9, pp. 90-98, 1972.

[10] J.P. Dauer, "On Controllability of Systems of the form $\dot{x} = A(t)x + g(t,u)$," J. Optimization Theory and Appl., vol. 11, pp. 132-138, 1973.

[11] E.N. Chukwu and S.D. Silliman, "Complete Controllability to a Closed Target Set," J. Optimization Theory and Appl., vol. 21, pp. 369-383, 1977.

[12] E.N. Chukwu and J.M. Gronski, "Controllability of Nonlinear Systems with Restrained Controls to Closed Convex Sets," Cleveland State University Report CSUMD 45, 1976.

STATE CONTROLLABILITY AND OBSERVABILITY FOR LINEAR NEUTRAL SYSTEMS*

DANIEL A. O'CONNOR
Electrical and Computer Engineering
Clarkson College of Technology
Potsdam, New York 13676

T. J. TARN
Systems Science and Mathematics
Washington University
St. Louis, Missouri 63130

ABSTRACT

The problem of state controllability and observability for linear neutral systems is studied. The approach is based on obtaining abstract evolution equations for the state of the system. Abstract conditions for approximate controllability are obtained and these conditions lead to algebraic conditions for a subclass of the systems considered. Finally observability conditions are obtained and shown to be "dual" to controllability conditions.

1. INTRODUCTION

In this paper we give necessary and sufficient conditions for approximate state controllability and observability of linear neutral systems. Our results are based on an abstract state space representation of the system in a Hilbert space.

It has been known for some time that linear retarded systems may be profitably investigated by the state space approach [1-3]. We believe that this paper is the first investigation of linear neutral systems from the state space point of view.

There is a certain freedom of choice in selecting the state space of our system. We choose to work in the Sobolev Space $W_{12}(-h,0:E^n)$. Other investigators who used this space are; Henry [4], in the theory of neutral functional differential equations, Banks, Jacobs [5], and Banks Jacobs and Langenhop [6], in the theory of optimal control and state controllability. Their work demonstrates that the space $W_{12}(-h,0:E^n)$ is specially suited for systems of neutral type.

The choice of this space forces us to consider abstract control systems with boundary control. Systems of this type arise often in the control theory of partial differential equations and various approaches to these problems has been developed. We use Fattorini's theory of boundary control systems [7], to develop state evolution equations for neutral systems.

Because of lack of space, we will only present the results and omit proofs, these may be found in [8].

We will adopt the following notations.

We take R to be the real line, C to be the complex plane, E^n the space of complex column n-vectors with norm $|x| = (\bar{x}^T x)^{1/2}$ and inner product $\langle x,y\rangle = x^T\bar{y}$, where x^T denotes transpose and $\bar{x}$ denotes complex conjugate.

The Sobolev space $W_{12}[-h,0:E^n] = W_{12}$ is the space of absolutely continuous functions ϕ: $[-h,0] \to E^n$, whose derivative, $\dot{\phi}$ is square integrable. If we endow the space W_{12} with the inner product $(\cdot\,|\,\cdot)$,

$$(\phi|\psi) = \phi^T(0)\bar{\psi}(0) + \int_{-h}^{0} \dot{\phi}^T(\theta)\bar{\dot{\psi}}(\theta)d\theta, \quad \phi,\psi \in W_{12},$$

then W_{12} is a Hilbert Space. We denote the norm induced by this inner product $||\cdot||$, i.e., $||\phi|| = (\phi|\phi)^{1/2}$ for $\phi \in W_{12}$.

---

*This work was supported partially by the National Science Foundation Grant Nos. ENG 75-09755 and ENG 76-80365.

$L_2[a,b:E^n]$ is the Lebesgue Space of equivalence classes of square integrable functions from $[a,b]$ into E^n.

$L_{2loc}[0,\infty:E^n]$ will represent the linear space of all equivalence classes of functions u, $u:[0,\infty] \rightarrow E^n$ such that u restricted to finite intervals is square integrable.

2. STATE SPACE REPRESENTATION

We consider systems of the form

$$\frac{d}{dt}\left[x(t) - \sum_{k=1}^{\ell} A_k x(t-h_k)\right] = \sum_{k=0}^{\ell} E_k x(t-h_k) + \int_{-h}^{0} E(\theta)x(t+\theta)d\theta + Bu(t) \qquad (1)$$

$$y(t) = Cx(t)$$

where $0 = h_0 < h_1 < h_2 < . . . < h_\ell = h$, A_k, E_k are real $n \times n$ matrices, B is an $n \times m$ real matrix, C is a $p \times n$ real matrix, $E(\cdot)$ is a real $n \times n$ matrix function on $[-h,0]$ with square integrable elements, $u(\cdot)$ is in $L_{2loc}[0,\infty:E^n]$.

If we prescribe some initial data $\phi \in W_{12}$ such that $\phi(t) = x(t)$ for $t \in [-h,0]$, then the initial value problem associated with (1) has a unique solution $x(t;\phi,u)$ on $t \geq -h$.

The state at time t; $x_t(\cdot;\phi,u)$ of the neutral functional differential equation (1) is a piece of function, $x_t(\cdot;\phi,u) : [-h,0] \rightarrow E^n$, constructed from the solution of the initial value problem as follows: for all $t \geq 0$, $\theta \in [-h,0]$

$$x_t(\cdot;\phi,u)\ (\theta) = \left\{\begin{array}{ll} x(t+\theta;\phi,u) & t+\theta>0 \\ \\ \phi(t+\theta) & t+\theta\leq 0 \end{array}\right\}.$$

The set of solution maps $\{T(t),\ t\geq 0\}$, where

$$T(t)\phi = x_t(\cdot;\phi,0)$$

is a strongly continuous semigroup of bounded linear operators on W_{12}. The infinitessimal generator of this semigroup $\tilde{A}$ is defined by

$$(\tilde{A}\psi)\ (\theta) = \dot{\psi}(\theta), \quad \theta \in [-h,0] \quad \text{for } \psi \in D(\tilde{A});$$

$$D(\tilde{A}) = \{\psi \in W_{12} | \dot{\psi} \in W_{12},\ \dot{\psi}(0) - \sum_{k=1}^{\ell} A_k\dot{\psi}(-h_k) - \sum_{k=0}^{\ell} E_k\psi(-h_k) - \int_{-h}^{0} E(\theta)\psi(\theta)d\theta = 0\}.$$

The characteristic matrix of system (1) is given by

$$\Delta(\lambda) = \lambda I - \sum_{k=1}^{\ell} A_k\lambda e^{-\lambda h_k} - \sum_{k=0}^{\ell} E_k e^{-\lambda h_k} - \int_{-h}^{0} E(\theta)e^{\lambda\theta}d\theta$$

and the spectrum of $\tilde{A}$, $\sigma(\tilde{A})$, is pure point spectrum consisting of the roots of det $\Delta(\lambda)$. The resolvent set of $\tilde{A}$ is denoted by $\rho(\tilde{A})$.

Associated with system (1) we define the abstract boundary control problem,

$$\begin{array}{lll} \frac{d}{dt} z(t) = \sigma z(t) & & t > 0 \\ \\ \tau z(t) = Bu(t) & & t \geq 0 \\ \\ z(0) = \phi & & \end{array} \qquad (2)$$

where

$\sigma\colon\ W_{12} \to W_{12}$ is the closed linear operator defined by

$$(\sigma\ \psi)\ (\theta) = \dot{\psi}(\theta)\ ,\quad \theta \in [-h,0]; \tag{3}$$

for all $\psi \in D(\sigma) = W_{22} = \{\phi \in W_{12} | \dot{\phi} \in W_{12}\}$.

$\tau\colon\ W_{12} \to E^n$ is a linear operator defined by

$$\tau\psi = \dot{\psi}(0) - \sum_{k=1}^{\ell} A_k \dot{\psi}(-h_k) - \sum_{k=0}^{\ell} E_k \psi(-h_k) - \int_{-h}^{0} E(\theta)\psi(\theta)d\theta; \tag{4}$$

for all $\psi \in D(\sigma) = W_{22}$.

B is the $n \times m$ constant matrix defined above, the initial data $\phi(\cdot)$ is the function specified in (1), and the control function u is identical to the function given in (1).

Note that if we compare the operators defined by (3) and (4) and the infinitessimal generator $\tilde{A}$ we observe that the restriction of σ to the null space of τ is the operator $\tilde{A}$. Moreover if the control u is identically zero system (2) is equivalent to the homogeneous equation

$$\frac{d}{dt}\, z(t) = \tilde{A}z(t),\quad t > 0,\quad z(0) = \phi$$

which has a unique solution $z(t) = T(t)\phi$, $t \geq 0$, for all $\phi \in W_{12}$.

A strong solution of the abstract boundary control problem in $t \geq 0$ is a W_{12} valued continuously differentiable function z(t) defined in $t \geq 0$ such that $z(t) \in W_{22}$ for all $t \geq 0$ and that (2) holds everywhere. The relationship between systems (1) and (2) is expressed in the following proposition.

Proposition 1. If u is twice continuously differentiable, $u(0) = 0$ and $\phi \in D(\tilde{A})$ then the state function $x_t(\cdot;\phi,u)$ corresponding to the solution $x(t;\phi,u)$ of (1) is a strong solution of the abstract boundary control problem (2).

The advantage of posing problem (1) as an abstract control problem is that in solving system (2) we obtain a representation for the state of our system which will be useful in developing conditions for approximate state controllability. We will now obtain a strong solution of (2) using techniques first proposed in [7].

We introduce the auxilliary boundary operator $\tilde{B}_\lambda | E^m \to W_{12}$ defined by

$$(\tilde{B}_\lambda u)\ (\theta) = e^{\lambda\theta}\Delta^{-1}(\lambda)Bu,\quad \theta[-h,0],\quad \text{for some } \lambda \in \rho(\tilde{A}).$$

The image of $\tilde{B}_\lambda$ is contained in W_{22}, and $\tau[\tilde{B}_\lambda\phi] = \Delta(\lambda)\Delta^{-1}(\lambda)Bu = Bu$. Assuming that the conditions of proposition 1 hold we look for a solution of (2) of the form,

$$z(t) = V(t) + \tilde{B}_\lambda u(t)$$

Substituting this expression in (2) gives us

$$\frac{d}{dt}\, V(t) = \sigma V(t) - \tilde{B}_\lambda u'(t) + \sigma\tilde{B}_\lambda u(t),\quad t > 0; \tag{5}$$

$$V(0) + \tilde{B}_\lambda u(0) = \phi; \tag{6}$$

$$\tau V(t) = 0,\quad t \geq 0. \tag{7}$$

Equations (5) - (7) are equivalent to

$$\frac{d}{dt}V(t) = \tilde{A}V(t) - \tilde{B}_\lambda u'(t) + \sigma\tilde{B}_\lambda u(t), \quad t \geq 0; \tag{8}$$

$$V(0) = \phi.$$

Since we assumed that u is twice continuously differentiable, (8) can be integrated and a strong solution of (2) is given by

$$z(t) = T(t)\phi + \int_0^t T(t-s)(\sigma\tilde{B}_\lambda u(s) - \tilde{B}_\lambda u'(s))ds + \tilde{B}_\lambda u(t). \tag{9}$$

We may simplify the integral formula (9). Notice that $(\sigma\tilde{B}_\lambda u)(\theta) = \lambda e^{\lambda\theta}\Delta^{-1}(\theta)Bu$, $\theta \in [-h,0]$ for all $u \in E^m$, therefore $\sigma\tilde{B}_\lambda = \lambda\tilde{B}_\lambda$. ($\lambda\tilde{B}_\lambda$ denotes scalar multiplication). Hence we may rewrite (9) as

$$z(t) = T(t)\phi + \lambda\int_0^t T(t-s)\tilde{B}_\lambda u(s)ds - \int_0^t T(t-s)\tilde{B}_\lambda u'(s)ds + \tilde{B}u(t).$$

Since u'(t) is assumed to be continuously differentiable, we can apply the integration by parts formular for semigroups to obtain

$$z(t) = T(t)\phi + (\lambda-\tilde{A})\int_0^t T(t-s)\tilde{B}_\lambda u(s)ds, \text{ for all } t \geq 0. \tag{10}$$

By proposition 1 and uniqueness of strong solutions of system (2), the state of system (1) at time t can be expressed by (10). The next proposition shows that this expression for the state is valid even if the smoothness assumptions in ϕ and u are dropped.

Proposition 2. For all $u \in L_{2,loc}[0,\infty:E^m]$ and $\phi \in W_{12}$,

$$x_t(\cdot;\phi,u) = T(t)\phi + (\lambda-\tilde{A})\int_0^t T(t-s)\tilde{B}_\lambda u(s)ds, \quad t \geq 0.$$

An abstract representation for the output of system (1) is given by

$$y(t) = \tilde{C}x_t(\cdot;\phi,u)$$

where $\tilde{C}|W_{12} \to E^p$, $\tilde{C}\phi = C\phi(0)$.

Based on these abstract relations for the state we may now formulate the state controllability and observability problems and obtain abstract conditions for these problems to have solutions.

3. STATE CONTROLLABILITY AND OBSERVABILITY CONDITIONS

Definition 1. The space of attainable states of system (1) is the linear manifold K_t in W_{12} defined by

$$K_t = \{x_t(\cdot;0,u) \mid u \in L_2[0,t_1:E^m]\}.$$

By the representation (10) we may also define K_t by

$$K_t = \{(\lambda-\tilde{A})\int_0^t T(t-\sigma)\tilde{B}_\lambda u(\sigma)d\sigma \mid u \in L_2[0,t;E^m]\},$$

we define the linear manifold, $K_\infty = \bigcup_{t>0} K_t$.

Definition 2. System (1) is approximately controllable if

$$\overline{K}_\infty = W_{12} \text{ (where } \overline{K}_\infty \text{ is the } W_{12}\text{-closure of } K_\infty).$$

Using standard techniques to obtain conditions for density of the subspace K_∞, [7] we obtain the following result.

Theorem 1. System (1) is approximately controllable if and only if for all $\phi \in W_{12}$,

$$[\phi^T(0) + \int_{-h}^{0} \lambda e^{\lambda\theta}\cdot\phi^T(\theta)\,d\theta]\Delta^{-1}(\lambda)B = 0 \tag{11}$$

for all $\lambda \in \rho(\tilde{A})$, implies $\phi \equiv 0$.

This condition for approximate controllability is similar to the conditions obtained in [1] for approximate controllability of linear retarded systems in the Hilbert space $M_{12}[-h,0:R^n]$. In [1] the conditions analogous to (11) lead to algebraic controllability conditions for retarded systems with commensurable delays. Similar although more complicated algebraic conditions can be obtained for neutral systems, this case will be considered in a forthcoming paper by the authors.

Definition 3. We say the observed system (1) is observable if $y(t) = 0$, $t \geq 0$ implies $x(t;\phi,0) = 0$ for $t \geq 0$.

Using Laplace transform method one can obtain the following equivalent condition.

Proposition 3. System (1) is observable if and only if for $\phi \in W_{12}$

$$(e^{\lambda\cdot}|F\,\phi)^T\Delta^{-T}(\lambda)C^T = 0 \quad \text{for all } \lambda \in \rho(\tilde{A}) \text{ implies } F\phi = 0;$$

where $e^{\lambda\cdot}$ represents the function $e^{\lambda\theta}$, $\theta \in [-h,0]$ and F is a bounded linear operator on W_{12}, characterized by

$$\begin{aligned}[F\psi](\theta) = \psi(0) &- \sum_{k=1}^{\ell} A_k\psi(-h_k) - \int_{-h_k}^{0} E_k\psi(-\theta-h_k)\,d\theta \\ &+ \int_{-h}^{0}\int_{-h}^{s} E(\theta)\psi(\theta-s)\,d\theta ds - \sum_{k=1}^{\ell}\int_{\min(-h_k,\theta)}^{0} A_k\psi(-s-h_k)\,ds \\ &+ \sum_{k=1}^{\ell}\int_{\min(-h_k,\theta)}^{0}\int_{-h_k}^{s}\psi(-p-h_k)\,dp\;ds \\ &- \int_{0}^{\theta}\int_{-h}^{s}\int_{-h}^{p} E(\sigma)\psi(\sigma-p)\,d\sigma\;dp\;ds \quad \text{for all } \psi\in W_{12},\ \theta\in[-h,0].\end{aligned} \tag{12}$$

F is called the hereditary operator associated with system (1). The bilinear form $(\psi|F\phi)$ on W_{12} is the Hale-type hereditary product, [9].

The formal adjoint system associated with (1) is given by

$$\frac{d}{dt}\left[x(t) - \sum_{k=1}^{\ell} A_k^T x(t-h_k)\right] = \sum_{k=0}^{\ell} E_k^T x(t-h_k) + \int_{-h}^{0} E^T(\theta)x(t+\theta)\,d\theta + C^T u(t) \tag{13}$$

we say that system (13) is F-approximately controllable [3] if

$$\overline{F\,K_\infty} = \overline{\mathrm{Im}\ (F)}$$

(Here F is the hereditary operator constructed from the coefficients, $A_k^T, E_k^T, E^T(\theta)$.) One can show that (13) is F-approximately controllable if and only if for $\phi \in W_{12}$

$$(e^{\lambda\cdot}|F^*\phi)^T\Delta^{-1}(\lambda)C^T = 0 \quad \text{for all } \lambda \in \rho(\tilde{A}) \text{ implies } F^*\phi = 0,$$

where F^* is the W_{12} - adjoint of F. It can be shown that F^* is of the same form as (12) with A_k, E_k, $E(\theta)$ replaced by their transpose. Thus we see that

Theorem 2. System (1) is observable if and only if system (13) is F-approximately controllable.

Theorem 2 is the analog of the duality between controllability and observability for finite dimensional linear dynamic systems. Versions of Theorem 2 for retarded systems were obtained earlier [1].

4. NECESSARY AND SUFFICIENT CONDITIONS FOR APPROXIMATE CONTROLLABILITY

In this section we improve the conditions for approximate state controllability given by (11). We will not consider state observability but similar results hold.

System (1) is said to be spectrally controllable if the closed subspace generated by the generalized eigenspaces of $\tilde{A}$, $M\sigma$, is contained in the closure of K_∞. As was shown in [10] (or can be derived from condition (11), [8]) system (1) is spectrally controllable if and only if

$$\operatorname{rank}\,[\Delta(\lambda),B] = n \quad \text{for all } \lambda \in \sigma(\tilde{A}).$$

This concept of controllability is weaker than approximate state controllability since the subspace generated by the generalized eigenspaces of $\tilde{A}$, need not be dense in W_{12}. However, for a special class of systems $\overline{M\sigma} = W_{12}$, systems with this property are said to be spectrally complete.

For example, consider the system of the form

$$\frac{d}{dt}\,[x(t) - A_{-1}x(t-h)] = A_0x(t) + A_1x(t-h) + bu(t). \tag{14}$$

It is proved in [6] that this system will be spectrally complete if $\det\,[\lambda A_{-1} + A_1]$ is not identically zero.

Even if this condition is not met it is possible that spectral controllability is equivalent to approximate state controllability. For example, let b be an $n \times 1$ matrix and (A_{-1},b) a controllable pair. There exists a change of basis for the system under which the pair (A_{-1},b) is transformed into standard controllable form. Thus without loss of generality we may take

$$A_{-1} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & 1 \\ a_0 & a_1 & \cdots & a_n \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

We note the following fact, first observed in [6].

For systems with scalar control and input matrix $b^T = (0,0,\ldots,1)$, approximate state controllability is independent of the elements of the last row of $\Delta(\lambda)$. This observation leads to the following theorem.

Theorem 3. If (A_{-1},b) is a controllable pair then system (14) is approximately controllable if and only if it is spectrally controllable.

5. EXAMPLE

Consider the transmission line equation

$$\frac{\partial v}{\partial z} = -L\,\frac{\partial i}{\partial t}$$

$$\frac{\partial i}{\partial z} = -C\,\frac{\partial v}{\partial t}\,, \quad 0 \le z \le 1\,, \quad t \ge 0$$

with boundary conditions

$$-v(0,t) = R_0 i(0,t) + L_0 \frac{di}{dt}(0,t) - E_0(t)$$

$$v(1,t) = R_1 i(1,t) + L_1 \frac{di}{dt}(0,t).$$

If we define the new variables

$$x_1(t) = \sqrt{C}\ \ v(0,t) + \sqrt{L}\ \ i(0,t)$$
$$x_2(t) = \sqrt{C}\ \ v(1,t) + \sqrt{L}\ \ i(1,t)$$

then we have

$$\begin{aligned} \frac{d}{dt}\,[x_1(t) + x_2(t-h)] &= \alpha_0 x_1(t) + \alpha_1 x_2(t-h) + E_0(t) \\ \frac{d}{dt}\,[x_2(t) + x_1(t-h)] &= \beta_0 x_2(t) + \beta_1 x_1(t-h) \end{aligned} \qquad (15)$$

where α_i, β_i $i=0,1$ are constants computed from R_i, L_i, L,C. The characteristic matrix of (15) is

$$\Delta(\lambda) = \begin{bmatrix} \lambda-\alpha_0 & (\lambda-\alpha_1)e^{-\lambda h} \\ (\lambda-\beta_1)e^{-\lambda h} & \lambda-\beta_0 \end{bmatrix}$$

and rank $[B, A_{-1}B] = 2$ thus (B, A_{-1}) is a controllable pair. Hence the system is approximately controllable if and only if it is spectrally controllable.

The system is spectrally controllable if and only if

$$\text{rank} \begin{bmatrix} \lambda-\alpha_0 & (\lambda-\alpha_1)e^{-\lambda h} & 1 \\ (\lambda-\beta_1)e^{-\lambda h} & \lambda-\beta_0 & 0 \end{bmatrix} = 2$$

hence it is spectrally controllable if and only if $\beta_1 \neq \beta_0$.

6. CONCLUSION

In this paper we have obtained state evolution equations for linear neutral systems in the space W_{12}. Based on the solutions of these equations we have obtained abstract conditions for approximate state controllability and observability. We have also noted the duality relationship between these two concepts. We remark that similar results can also be obtained for systems with delay in control [8].

Since the conditions expressed in (11) is not yet in a usable form we have investigated the relationship between state approximate controllability and a related concept, spectral controllability. For a class of systems with scalar control we have given sufficient conditions for approximate state controllability and spectral controllability to be equivalent. In this case approximate state controllability is characterized by the convenient condition,

$$\text{rank} \quad [\Delta(\lambda), B] = n \qquad \text{all } \lambda \in \sigma(\tilde{A}).$$

REFERENCES

1. A. Manitius and R. Triggiani, "Function Space Controllability of Linear Retarded Systems: A Derivation from Abstract Operator Conditions," SIAM Journal on Control and Optimization, 16, No. 4. (1978).

2. A. Manitius and R. Triggiani, Sufficient Conditions for Function Space Controllability and Feedback Stabilization of Linear Retarded Systems," Proceedings of the 1976 IEEE Conference on Decision and Control.

3. A. Manitius, "Controllability, Observability and Stability of Retarded Systems," Proceedings of 1976 IEEE Conference on Decision and Control.

4. D. Henry, "Linear Autonomous Functional Differential Equations in the Sobolev Space $W_2^{(1)}$, Technical Report, Department of Mathematics, University of Kentucky, Lexington, Kentucky.

5. H. T. Banks, M. Q. Jacobs and C. E. Langenhop, "Characterization of the Controlled States in $W_2^{(1)}$ of Linear Hereditary Systems," SIAM Journal on Control, 13 (1975) pp. 611-649.

6. M. Q. Jacobs and C. E. Langenhop, "Criteria for Function Space Controllability of Linear Neutral Systems," SIAM Journal on Control and Optimization, 14 (1976) pp. 1009-1048.

7. H. Fattorini, "Boundary Control Systems," SIAM Journal on Control, Vol. 6, No. 3, 1968.

8. D. O'Connor, "State Controllability and Observability of Linear Neutral Systems" D.Sc. dissertation, Washington University, 1978.

9. J. Hale, Theory of Functional Differential Equations, Springer-Verlag, New York, 1977.

10. L. Pandolfi, "Stabilization of Neutral Functional Differential Equations," Journal of Optimization Theory and Applications 20 (1976) 191-204.

ON THE INDUCED SUBFLOW IN REGULATOR SYSTEMS ON DIFFERENTIABLE MANIFOLDS

J S A HEPBURN & W M WONHAM

Systems Control Group,
Dept of Electrical Engineering,
University of Toronto,
Toronto, Canada, M5S 1A4.

ABSTRACT

In the analysis of the regulator problem the tracking signals may be regarded as being generated by an *exosystem*, or dynamic model of the *outside world*. The dynamic behavior of this exosystem induces a subflow in the overall regulator dynamics. Conditions are presented in the setting of differentiable manifolds for the existence, uniqueness and property of global stable attraction of the subflow.

INTRODUCTION

In recent years it has become widely recognised that geometric concepts play an important role in the analysis and understanding of control systems. While the theory has been developed extensively for linear multivariable systems [1], the extension of the ideas to nonlinear systems is still in its infancy, but research in this direction appears fruitful.

The Internal Model Principle (IMP) for regulators has been established in the framework of linear multivariable systems [2] and a class of abstract automata [4], and certain implications have been explored in a nonlinear setting [3]. In these analyses it is apparent that a subflow is induced in the overall regulator system dynamics by the exogenous signals which the regulator is required to track. As this subflow clearly plays a central role in the study of such regulators, knowledge of its properties is important; it is the purpose of this paper to present, in the setting of differentiable manifolds, criteria guaranteeing the existence and uniqueness of the subflow as an embedding into the state space, together with the property of being a stable attractor for all states - thus generalising some results from the linear case.

In this treatment, the exogenous signals are regarded as being generated by an *exosystem* represented by the pair (X_e, ξ_e), where X_e is a differentiable manifold having the role of a state set and $\xi_e : X_e \to TX_e$ a vector field representing the dynamic behavior of the exosystem. The regulator itself comprises a plant and controller, and it is convenient to regard it, together with the exosystem, as a composite system $(X,\xi) = (X_s \times X_e, (\xi_s,\xi_e))$ where X_s and ξ_s specify the state and dynamics of the controller and plant.

The tracking requirement is specified *a priori* as a relation between the states of the plant and exosystem. With the adoption of the product structure above, this relation may be embodied in a submanifold K of X. The task of the controller is to make the plant state satisfy this relation, in other words to bring the overall system state into K. Typically this is accomplished gradually, corresponding to asymptotically zero tracking error in the linear case. As a simple example, if $X_s \times X_e = \mathbb{R}^2 \times \mathbb{R}$ and the plant is required to track the exosystem exactly - i.e. it is desired that $x_p = x_e$, where x_p is the plant state, x_e the state of the exosystem and x_c the controller state with $x_s = (x_p,x_c)$ - then K would be the 2-dimensional subspace of $\mathbb{R}^3$:

$$K = \{(x_p,x_c,x_e) \mid x_p = x_e\}$$

The exosystem induces a certain subflow in the composite system (X,ξ). Ideally the regulator should be *tractable* to the exosystem in the sense that this subflow should copy the flow of the exosystem; more precisely, there is an embedding $\nu\colon X_e \to X$ such that the following diagram commutes:

$$\begin{tikzcd}
X_e \arrow[r, "\xi_e"] \arrow[d, "\nu"] \arrow[dd, bend right=60, "\mathrm{id}"'] & TX_e \arrow[d, "\nu_*"] \arrow[dd, bend left=60, "\mathrm{id}"] \\
X_s \times X_e \arrow[r, "\xi"] \arrow[d, "pr_2"] & TX_s \times TX_e \arrow[d, "pr_2"] \\
X_e \arrow[r, "\xi_e"] & TX_e
\end{tikzcd} \tag{1.1}$$

If this is the case then the regulation requirement may be expressed as

(i) $\nu(X_e)$ is a global attractor for all states

(ii) $\nu(X_e) \subset K$ (1.2)

These two conditions ensure that regardless of the initial state, the regulator will ultimately perform the tracking task arbitrarily well.

A *system* is a pair (X,ξ), where X is a Hausdorff C^∞ differentiable manifold of dimension n and $\xi: X \to TX$ is a C^∞ vector field. The notation follows Lang [5]. A Riemannian structure on X is an assignment, varying smoothly with x, of an inner product to each tangent space T_xX of X. This structure induces on X a unique function called the Riemannian metric.

For each x there is a unique maximal integral curve $\phi_x: J_x \to X$ satisfying

$$\phi_x'(t) = \xi(\phi_x(t))$$
$$\phi_x(0) = x$$

where $J_x \subset \mathbb{R}$. If for each x, $J_x = \mathbb{R}$ then ξ is *complete*. This will be a standing assumption. The *flow* of ξ is the map

$$\phi: \mathbb{R} \times X \to X$$
$$(t,x) \mapsto \phi_x(t)$$

For simplicity, the notation $x(t)$ will sometimes be used instead of $\phi(t,x)$.

The Lie derivative of a C^∞ function $V: X \to \mathbb{R}$ with respect to ξ is the C^∞ function $L_\xi V$ on X, and is locally given by

$$L_\xi V(x) = DV(x)\xi(x)$$

A *critical element* of the flow ϕ of ξ is a fixed point or closed orbit. Let

$$C(\xi) = \{x \mid x \in \gamma,\ \gamma \text{ a critical element of } \phi \text{ of } \xi\}$$
$$C_\alpha(\xi) = \{x \mid \text{the negative limit set of } x \text{ is contained in } C(\xi)\}$$

Clearly for each critical element γ of ϕ there is $\tau_\gamma > 0$ such that for each $x \in \gamma$

$$\phi(\tau_\gamma, x) = x$$

Each critical element γ has a pseudochart (ψ,U) where U is an open nbhd of γ, such that the principal part of the local representative of ξ with respect to (ψ,U) satisfies

$$\xi_{\psi_1}(\theta,u) = 1 + Q(\theta,u)$$
$$\xi_{\psi_2}(\theta,u) = Au + R(\theta,u)$$

where $\xi_{\psi_1} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}$ and $\xi_{\psi_2} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ ($m + 1 = n$, the dimension of X if γ is a closed orbit, while if γ is a critical point then the first equation and the dependence on θ may be dropped so that $m = n$), A is a linear map and Q and R are periodic in θ and vanish together with their partial derivatives with respect to u for $u = 0$. (See Abraham & Robbin, §22 and

§26 [6].

The spectrum of a linear map $A\colon \mathbb{R}^n \to \mathbb{R}^n$ will be denoted by $\sigma(A)$, and for two linear maps A and B, the ordering

$$\sigma(A) < \sigma(B) < \alpha$$

will indicate that the real parts of the eigenvalues of A are less than those of B, all of which are less than the real number α.

THE INDUCED SUBFLOW

With X having the product structure $X_s \times X_e$, the vector field ξ has the form (ξ_s,ξ_e) with

$$\begin{aligned} \xi_s &\colon X_s \times X_e \to TX_s \\ \xi_e &\colon X_e \to TX_e \end{aligned} \tag{3.1}$$

expressing the fact that the exosystem is not influenced by the regulator. The flow of ξ will have the form $\phi = (\phi_s,\phi_e)$ with

$$\begin{aligned} \phi_s &\colon \mathbb{R} \times X_s \times X_e \to X_s \\ \phi_e &\colon \mathbb{R} \times X_e \to X_e \end{aligned} \tag{3.2}$$

The purpose of this section is to show first that under certain conditions there is a unique embedding $\nu_c\colon C(\xi_e) \to X$ satisfying

$$pr_2 \circ \nu_c = id_{C(\xi_e)} \tag{3.3a}$$

$$\nu_{c*} \circ \xi_e|C(\xi_e) = \xi \circ \nu_c \tag{3.3b}$$

In other words, ν_c satisfies (1.1) with $C(\xi_e)$ replacing X_e. Then with further assumptions ν_c may be extended uniquely to an embedding ν satisfying (1.1). Finally, under these conditions $\nu(X_e)$ is a global attractor for the flow ϕ of ξ.

Under the assumption that X has a Riemannian structure, so that there is a Riemannian metric ρ, a Lyapunov-like condition is used to guarantee that the distance (with respect to ρ) between any two points (x_s,x_e) and (x'_s,x_e) in X tends to zero exponentially fast.

Theorem 1:

Let X have a Riemannian structure and corresponding metric $\rho\colon X \times X \to \mathbb{R}^+$. Suppose there are a C^∞ function

$$V\colon X_s \times X_s \times X_e \to \mathbb{R}^+$$

satisfying for each (x_s,x'_s,x_e)

(i) $V(x_s,x'_s,x_e) \geq \rho^2((x_s,x_e),(x'_s,x_e))$

(ii) $V(x_s,x_s,x_e) = 0$

and a number $\varepsilon > 0$ such that the Lie derivative of V w. r. t. $\hat{\xi}$ satisfies

$$L_{\hat{\xi}} V \leqq -\varepsilon V$$

where

$$\hat{\xi}: X_s \times X_s \times X_e \to TX_s \times TX_s \times TX_e$$

$$(x_s,x'_s,x_e) \mapsto (\xi_s(x_s,x_e),\xi_s(x'_s,x_e),\xi_e(x_e))$$

Then

(a) for all (x_s,x_e), (x'_s,x_e)

$$\rho((x_s(t),x_e(t)),(x'_s(t),x_e(t))) \downarrow 0$$

where

$$x_s(t) := \phi_s(t,(x_s,x_e)) \quad \text{etc.}$$

(b) there is a unique embedding

$$\nu_c : C(\xi_e) \to X_s \times X_e$$

satisfying (3.3).

Proof:

(a) Suppose $L_{\hat{\xi}} V \leqq -\varepsilon V$. Then for each (x_s,x'_s,x_e)

$$V(x_s(t),x'_s(t),x_e(t)) \leqq V(x_s,x'_s,x_e)e^{-\varepsilon t}$$

and

$$\rho[(x_s(t),x_e(t)),(x'_s(t),x_e(t))] \leqq V^{\frac{1}{2}}(x_s,x'_s,x_e)e^{-\varepsilon t/2} \downarrow 0$$

(b) Let $\gamma \subset C(\xi_e)$ be a critical element of ϕ_e. For any $x_s \in X_s$ and $x_e \in \gamma$ form the sequence

$$\langle(x_i,x_e)\rangle_{i=0,1,2,...}$$

where $x_{i+1} = \phi_s(\tau_\gamma;(x_i,x_e))$ and $x_0 = x_s$

It is easy to show that

$$\rho((x_s,x_e),(x_n,x_e)) \leqq V^{\frac{1}{2}}(x_s,x_1,x_e)/(1 - \exp(-\varepsilon\tau_\gamma/2)) =: r\text{,say}$$

and

$$\rho((x_m,x_e),(x_n,x_e)) \leqq r|[\exp(-m\varepsilon\tau_\gamma/2) - \exp(-n\varepsilon\tau_\gamma/2)]|$$

Thus the sequence is bounded and Cauchy and has a limit, $\hat{x}_s$ say. From (a) above it is clear that for each x_e, $\hat{x}_s = \hat{x}_s(x_e)$ is independent of the choice of x_s, and hence unique. This holds for every $x_e \in C(\xi_e)$. Thus there is a unique map

$$\nu_c: C(\xi_e) \to X_s \times X_e$$

$$x_e \mapsto (\hat{x}_s(x_e),x_e)$$

Clearly ν_c satisfies (3.3a). For (3.3b) it is sufficient to show that for each $x_e \in C(\xi_e)$

$$\nu_c \circ \phi_e(t,x_e) = \phi(t,\nu_c(x_e))$$

But for some $\tau > 0$

$$\phi_e(\tau,x_e) = x_e$$

so that

$$\phi(\tau;\hat{x}_s,x_e) = (\hat{x}_s,x_e)$$

and for each t with $0 \leqq t \leqq \tau$

$$\phi(t+\tau;\hat{x}_s,x_e) = \phi(t;\hat{x}_s,x_e)$$
$$\phi_e(t+\tau;x_e) = \phi_e(t,x_e)$$

Since ν_c is unique, it follows that

$$\begin{aligned}\nu_c \circ \phi_e(t,x_e) &= \phi(t,(\hat{x}_s,x_e)) \\ &= \phi(t,\nu_c(x_e))\end{aligned}$$

Clearly ν_c is an injective immersion, and since its inverse image is a projection, it follows that ν_c is proper and hence an embedding. ∎

From the above, it may be noted that if Theorem 1 holds then γ is a critical element of ϕ_e iff $\nu_c(\gamma)$ is a critical element of ϕ. Before proceeding to the theorem extending ν_c to an embedding $\nu: X_e \to X$, some additional results are required.

Lemma:

Under the assumptions of Theorem 1, at each critical element $\nu_c(\gamma)$ of ϕ, where γ is a critical element of ϕ_e, there is a pseudochart (ψ_γ,U_γ) such that ξ has a local representation of the form

$$\begin{aligned}\xi_\theta &= a + \Theta(\theta,x_e) \\ \xi_{x_s} &= Ax_s + X_s(\theta,x_s,x_e) \\ \xi_{x_e} &= Bx_e + X_e(\theta,x_e)\end{aligned}$$

with A and B constant matrices in real canonical form and $\sigma(A) < 0$; X_s and X_e vanish together with their partial derivatives with respect to x_s and x_e when $(x_s,x_e) = 0$ and are periodic in θ.

Proof:

The existence of the local representation is standard [6]. The dependence of X_e and Θ on only θ and x_e follows from the form of ξ in (3.1). The fact that $\sigma(A) < 0$ results from Theorem 1. ∎

Theorem 2:

Given the real C^{∞} system of ordinary differential equations

$$\begin{aligned} \dot{\theta} &= a + \Theta(\theta,y) \\ \dot{x} &= Ax + X(\theta,x,y) \qquad (3.4) \\ \dot{y} &= By + Y(\theta,y) \end{aligned}$$

where Θ, X, Y vanish together with their partial derivatives w. r. t. x and y on the set $Z = \{(\theta,x,y) | \theta \text{ arbitrary}, (x,y) = 0\}$; $\sigma(A) < -\alpha < \sigma(B)$, $\alpha > 0$. Then for some β, $\delta > 0$ sufficiently small

(i) $\sigma(A) < -\alpha - \beta < -\alpha + \beta < \sigma(B)$

(ii) there exists an invariant integral manifold M_s

$$M_s = \{(\theta,x,y) | \theta \text{ arbitrary}, |x| < \delta, y = f(\theta,x)\}$$

with the property that any solution $\zeta(t)$ of (3.4) starting in M_s satisfies

$$\limsup_{t\to\infty} (\ln|\zeta(t)|)/t < -\alpha - \beta$$

and for some $\eta > 0$, any solution $\zeta(t)$, not in M_s, with $|\zeta(t)| < \eta$ for $t \geq 0$ satisfies

$$\liminf_{t\to\infty} (\ln|\zeta(t)|)/t > -\alpha + \beta$$

(iii) there exists an invariant integral manifold M_c

$$M_c = \{(\theta,x,y) | \theta \text{ arbitrary}, |y| < \delta, x = g(\theta,y)\}$$

(iv) $M_s \cap M_c = Z$, $M_s \pitchfork M_c$. M_s is locally unique,

(v) any solution of (3.4) tending to Z as $t \to \infty$, but not in M_s, is tangent to M_c in the limit as $t \to \infty$,

where f and g together with their partial derivatives w. r. t. x and y vanish on Z.

Proof:

(i) Trivial.

(ii) This is essentially Theorem 4.4, Chapter 13 of Coddington [7]. In this case, $f(\theta,x) = 0$ for $|x| < \delta$.

(iii) The method of proof will be briefly outlined. The nonlinear terms of the right of (3.4) are multiplied by a bump function with support contained in the ball $|(x,y)| < \delta$. M_c is determined by constructing the unique solution to the following differential-integral system (see Appendix C [6]):

$$\begin{aligned} \dot{\theta} &= a + \tilde{\Theta}(\theta,y) \\ g(\theta,y) &= \int_{-\infty}^{0} e^{-At}\tilde{X}(r(t),g(r(t),s(t)),s(t))dt \\ \dot{y} &= By + \tilde{Y}(\theta,y) \end{aligned}$$

where $\tilde{\Theta}$, $\tilde{X}$ and $\tilde{Y}$ are the modified nonlinear terms, and $r(t)$ and $s(t)$ are the unique solutions to the first and third equations and $(r(0),s(0)) = (\theta,y)$.

(iv) Clear.

(v) Let $\zeta(t) = (\theta(t),x(t),y(t))$ be a solution of (3.4) with $\zeta(t) \to Z$ as $t \to \infty$, and $|\zeta(t)| < \delta$ for $t \geqq 0$, and $y(t) \neq 0$. Let $x'(t) = g(\theta(t),y(t))$. Then by (ii) above, there are T, K, L > 0 such that for $t \geqq T$

$$|x(t) - x'(t)| \leqq K\exp[-(\alpha+\beta)t] \quad \text{and} \quad |y(t)| \geqq L\exp[-(\alpha-\beta)t]$$

so that

$$|x(t) - x'(t)|/|y(t)| \leqq (K/L)\exp(-2\beta t) \downarrow 0 \qquad \blacksquare$$

With these results it may now be shown that ν_c may be extended uniquely to a map satisfying (1.1).

Theorem 3:

Suppose Theorem 1 holds, so that ν_c exists and is unique, and that $X_e = C_\alpha(\xi_e)$, i.e. every point x_e of X_e has as its negative limit set a critical element of ϕ_e. Suppose that for each critical element γ of ϕ_e, Theorem 2 holds for the pseudochart of $\nu_c(\gamma)$ given by the Lemma, where x and y correspond to x_s and x_e respectively. Then there is a unique embedding

$$\nu\colon X_e \to X$$

extending ν_c. Furthermore ν satisfies (1.1) and $\nu(X_e)$ is a global stable attractor for the flow ϕ of ξ.

Proof:

By Theorem 2, in a nbhd of each critical element $\nu_c(\gamma)$ of ϕ there is a local invariant manifold $M_c(\gamma)$ given by

$$\nu_\gamma\colon N(\gamma) \to X$$

where $N(\gamma)$ is a nbhd of γ and ν_γ corresponds to g of Theorem 2. Since by hypothesis $X_e = C_\alpha(\xi_e)$, for each $x_e \in X_e$, there is a critical element γ such that $\alpha(x_e) = \gamma$. Furthermore there is $t_{x_e} \leqq 0$ such that for all $t \leqq t_{x_e}$

$$\phi_e(t,x_e) \in N(\gamma)$$

Define

$$\nu(x_e) = \phi(-t_{x_e},\nu_\gamma[\phi_e(t_{x_e},x_e)])$$

Then

$$\nu\colon X_e \to X_s \times X_e$$

extends ν_c and satisfies (1.1). For any $(x_s,x'_s,x_e) \in X_s \times X_s \times X_e$ and $t \leqq 0$,

$$V(x_s(t),x'_s(t),x_e(t)) \geqq V(x_s,x'_s,x_e)\exp(-\varepsilon t)$$

Thus if $x_s \neq x'_s$ then as $t \to -\infty$, $V \not\to 0$ so that $x_s(t) \not\to x'_s(t)$. Hence for each

x_e there is at most one x_s such that $(x_s,x_e)(t) \to \nu(\gamma)$ for some γ. Since ν exists, it is unique. To show that $\nu(X_e)$ is a global attractor for ϕ, take any $(x_s,x_e) \in X$. Then

$$\rho((x_s(t),x_e(t)),\nu(X_e)) \leqq \rho((x_s(t),x_e(t)),\nu(x_e(t)))$$
$$\leqq V^{\frac{1}{2}}(x_s,x_s',x_e)\exp(-\varepsilon t/2) \downarrow 0$$

where $(x_s',x_e) := \nu(x_e)$.

The spectral condition (viz. $\sigma(A) < \sigma(B)$) in Theorem 2 guarantees by Theorem 2(v) that any integral curve with some critical element as its positive limit set will approach that $\nu_c(\gamma)$ tangent to $M_c(\gamma)$ in the limit since it cannot be in $M_s(\gamma)$. This ensures that ν will be differentiable at critical elements. ∎

EXAMPLES

As an illustration of Theorem 1 consider the case where $X_s \times X_e = \mathbb{R}^m \times \mathbb{R}^n$, ρ is the Euclidean distance and

$$V(x_s,x_s',x_e) = \rho^2((x_s,x_e),(x_s',x_e)) = (x_s - x_s')^T(x_s - x_s')$$

Let

$$\dot{x}_s = f(x_s,x_e)$$
$$\dot{x}_e = g(x_e)$$

Then the Lie derivative of V for this system is

$$\frac{d}{dt}V(x_s(t),x_s'(t),x_e(t))$$
$$= [2(x_s - x_s')^T \quad -2(x_s - x_s') \quad 0][f^T(x_s,x_e) \quad f^T(x_s',x_e) \quad g^T(x_e)]^T$$
$$= 2(x_s - x_s')^T[f(x_s,x_e) - f(x_s',x_e)]$$
$$= (x_s - x_s')^T\int_0^1[D_1f(\theta x_s + (1-\theta)x_s',x_e) + D_1f^T(..)]d\theta(x_s - x_s')$$

Hence if $\sigma(D_1f(x_s,x_e) + D_1f^T(x_s,x_e)) \leqq -\varepsilon < 0$ then $L_{\hat{\xi}}V \leqq -\varepsilon V$. In particular if

$$f(0,0) = 0 \qquad g(0) = 0$$

then setting $x_s' = 0$, $x_e = 0$, the above shows that the origin is a global stable attractor for all x_s, which is just Krasovskii's theorem [8].

Finally, since the regulator structure has been analysed in the case of an exosystem producing step signals [3], it is of interest to apply the results of this paper to that situation. First, $\xi_e \equiv 0$ so that every x_e in X_e is a critical point and $X_e = C(\xi_e) = C_\alpha(\xi_e)$. Thus, provided Theorem 1 holds, the other theorems will also hold and $\nu = \nu_c$.

REFERENCES

1. W M Wonham: *Linear Multivariable Control: A Geometric Approach*, vol 101, Lecture Notes in Economics and Mathematical Systems (Springer-Verlag, 1974)
2. B A Francis & W M Wonham: Internal Model Principle of Control Theory, *Automatica*, vol 12, 457-465, 1976
3. W M Wonham: On Structurally Stable Nonlinear Regulation with Step Inputs, *20th Midwest Symp. on Circuits & Systems* (1977)
4. W M Wonham: Towards an Abstract Internal Model Principle, *IEEE Trans. Syst. Man & Cyb.*, vol SMC-6, 735-740, 1976
5. S Lang: *Differentiable Manifolds*, (Addison-Wesley, 1972)
6. R Abraham & J Robbin: *Transversal Mappings and Flows*, (W A Benjamin, 1967)
7. E A Coddington & N Levinson: *Theory of Ordinary Differential Equations*, (McGraw-Hill, 1955)
8. N N Krasovskii: On Stability with Large Perturbations (in Russian), *Prikl. Mat. Mekh.*, vol 21, 309-319, 1957

ON STABILIZATION OF CONSERVATIVE DYNAMICAL SYSTEMS

R. K. MILLER
Department of Mathematics
Iowa State University
Ames, Iowa 50011

A. N. MICHEL
Department of Electrical Engineering and
Engineering Research Institute
Iowa State University
Ames, Iowa 50011

ABSTRACT

In the present paper, necessary and sufficient conditions are established in order that the addition of damping (at random locations) to a conservative dynamical system ensures the asymptotic stability of the equilibrium. The present results have potential applications in control theory and they provide further insight into the mechanisms of stabilization.

I. INTRODUCTION

Consider the linear mechanical mass-spring system of Fig. 1 which is governed by the equations

$$\begin{aligned} m_1\ddot{x}_1 + k_1x_1 + k(x_1 - x_2) &= 0 \\ m_2\ddot{x}_2 + k_2x_2 + k(x_2 - x_1) &= 0, \end{aligned} \tag{1}$$

where x_i denotes the displacement of mass m_i and k_1, k_2, k denote linear spring constants. When the initial state of this conservative system is displaced from its equilibrium position, the system will remain in motion indefinitely. If linear viscous damping is added at some or all of the masses and springs, as shown in Fig. 2, then the governing equations become

$$\begin{aligned} m_1\ddot{x}_1 + k_1x_1 + k(x_1 - x_2) + B_1\dot{x}_1 + B(\dot{x}_1 - \dot{x}_2) &= 0 \\ m_2\ddot{x}_2 + k_2x_2 + k(x_2 - x_1) + B_2\dot{x}_2 + B(\dot{x}_2 - \dot{x}_1) &= 0 \end{aligned} \tag{2}$$

where $B_1 \geq 0$, $B_2 \geq 0$, $B \geq 0$ and $B_1 + B_2 + B > 0$. At a first glance, it would seem that the indiscriminate or random addition of such damping terms will stabilize the rest position, making system (2) asymptotically stable. Indeed, one could argue that since the addition of a dash pot at even one single location shown in Fig. 2 will reduce the total energy of the system, eventually all of the energy will be dissipated and the motion of the system will tend to its equilibrium.

The preceding argument is simple, appealing but unfortunately wrong. While for most values of the parameters the above conjecture is correct, it is not true when $B_1 = B_2 = 0$, $B > 0$, $k_1/m_1 = k_2/m_2$, for in this case the two masses can be made to move in synchronism. When this happens, $x - x_2$ is constant, the term $B(\dot{x}_1 - \dot{x}_2)$ has no effect on the motion and no dissipation of energy will occur.

The conservative system (1) and its corresponding damped system (2) are simple enough to be analyzed by simple inspection. However, in the case of general, complex, stable linear conservative systems, it is far from trivial to decide where damping should be added in order to ensure that the rest position will be asymptotically stable. Similar questions can be asked with respect to adding dissipative terms in electrical systems, electromechanical sysems, and so forth.

In the present paper we give necessary and sufficient conditions which answer the questions raised above for linear systems. Although we give results involving several cases, our main result answers the following question: for a stable and conservative system, what are necessary and sufficient conditions which ensure that an associated damped system will be asymptotically stable?

II. GENERAL LINEAR SYSTEMS

We will employ the following frequently used definition.

Definition. Let U and V be matrices of dimensions $m \times n$ and $n \times n$, respectively. We say that the pair (U,V) is observable if and only if the matrix

$$\begin{bmatrix} U \\ UV \\ UV^2 \\ \cdots\cdots \\ UV^{n-1} \end{bmatrix}$$

has full rank. ■

We consider a linear system of differential equations given by

$$\dot{x} = Ax \tag{3}$$

where $x \in R^n$, $t \in J = [t_0, \infty)$, $t_0 \geq 0$, $\dot{x} = dx/dt$, and A is an $n \times n$ matrix. We let $x(t;\, x_0,\, t_0)$ denote the solutions of (3) with $x_0 = x(t_0;\, x_0, t_0)$. We assume that the trivial solution $x \equiv 0$ of (3) is stable (in the Lyapunov sense (see Hahn [3])) so that there is a positive definite matrix G (i.e., $G > 0$) such that the matrix

$$B = A^T G + GA$$

is negative semidefinite (i.e., $B \leq 0$). Thus, there exists a Lyapunov function $v\colon\ R^n \to R$ with

$$v(x) = x^T G x$$
$$Dv_{(3)}(x) = x^T B x \tag{4}$$
$$B = A^T G + GA,$$

where $Dv_{(3)}(x)$ denotes the derivative of v with respect to t along the solutions of (3). Our first result is as follows.

Theorem 1. For system (3) assume a Lyapunov function (4) such that $G > 0$ and $B \leq 0$. Then the trivial solution of system (3) is asymptotically stable (in the Lyapunov sense (see [3])) if and only if the pair (B,A) is observable.

Proof. Suppose that (B,A) is observable and let

$$N = \{x : Bx = 0\} = \{x : Dv_{(3)}(x) = 0\}.$$

Let N_I be the largest subset of N which is invariant with respect to system (3). (I.e., N_I is the largest subset of N such that $x_0 \in N_I$ implies $x(t;x_0,t_0) \triangleq x(t) \in N_I$ for all $t \in R$.) Now if $x(t) \in N_I$, then $Bx(t) \equiv 0$ so that

$$0 = \frac{d}{dt}(Bx) = B\dot{x} = BAx$$
$$0 = \frac{d}{dt}(BAx) = BA\dot{x} = BA^2x$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$0 = \frac{d}{dt}(BA^{n-2}x) = BA^{n-2}\dot{x} = BA^{n-1}x.$$

Since (B,A) is observable, we must conclude that any trajectory x(t) in N_I is the rest state, i.e., $N_I = \{0\}$. By the invariance principle (see, e.g., [1,4,5]) it follows that $x \equiv 0$ is asymptotically stable.

Conversely, assume that (B,A) is not observable. Then there is a trajectory $x(t) = [\exp(At)]x_0, x_0 \neq 0$, such that $B[\exp(At)]x_0 \equiv 0$. For this trajectory,

$$\frac{d}{dt} v(x(t)) = x^T(t)Bx(t) = x^T(t) \cdot 0 = 0,$$

and so $v(x(t)) \equiv v(x_0) > 0$ for all $t \geq 0$. Thus, x(t) can not tend to the origin as $t \to \infty$, i.e., the trivial solution $x \equiv 0$ is not asymptotically stable.■

The method of proof used in Theorem 1 can be modified to establish the following more general result.

Theorem 2. For system (3) assume that (4) is true with $G > 0$ and $B \leq 0$. Define

$$N = \{x : Dv_{(3)}(x) = 0\}.$$

Suppose that there exists a matrix C such that the set

$$N_1 \triangleq \{x : Cx = 0\}$$

equals N, and suppose there exists a matrix D such that

$$N_2 \triangleq \{x: Dx = 0\} \supset N_1.$$

Then the trivial solution of (3) is asymptotically stable if and only if the pair (C,A-D) is observable.

Proof. First we consider a trajectory $x(t) \in N$ for $-\infty < t < \infty$. Then $Cx(t) \equiv 0$ so that $Dx(t) \equiv 0$ and

$$0 = \frac{d}{dt}(Cx) = C\dot{x} = CAx = C(A-D)x,$$

$$0 = \frac{d}{dt}[C(A-D)x] = C(A-D)\dot{x} = C(A-D)^2 x,$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$0 = C(A-D)^{n-1}x.$$

Since (C,A-D) is observable, it follows that $x(t) \equiv 0$. By the invariance result in [1] it follows that the equilibrium $x \equiv 0$ is asymptotically stable.

Now suppose that $N_2 \supset N_1$ and that (C,A-D) is not observable. Then there exists $x_0 \neq 0$ such that $C\{\exp[(A-D)t]\}x_0 \equiv 0$. Since $N_1 = N$ it follows that

$$\frac{d}{dt}v(x(t)) = x^T(t)Bx(t) \equiv 0, \qquad x(t) \triangleq \exp[(A-D)t]x_0,$$

and

$$v(x(t)) \equiv v(x_0) > 0.$$

Thus, x(t) cannot tend to the origin. Also, since $N_2 \supset N_1$, we have

$$\dot{x}(t) = (A-D)x(t) = Ax(t) - 0 = Ax(t),$$

i.e., x(t) solves Eq. (3). Thus, the equilibrium of (3) is not asymptotically stable. ■

For Theorem 2 there are many possible choices for the matrices C and D. For example, $C = B$ and either $D = 0$ or $D = \pm B$ will do. As another example, since B is symmetric and negative semidefinite, there exists a matrix C such that $C^*C = -B$. For this choice of C we must choose $D = 0$, or $D = \pm C$, or $D = \pm B$. For a third way of choosing the matrices C and D, refer to Theorem 3 in the next section.

We note that Professor David Russell [6] has communicated to the authors a version of Theorem 2 with $C^*C = -B$ and $D = 0$

III. HAMILTONIAN SYSTEMS

The Hamiltonian formulation in mechanics is well established (see e.g., [2,7]). Given a Hamiltonian function $H(q_1,\ldots, q_n, p_1,\ldots, p_n)$, the Hamiltonian differential equations for conservative systems are

$$\dot{q}_i = \frac{\partial H}{\partial p_i}(q_1,\ldots, q_n, p_1,\ldots, p_n)$$

$$\dot{p}_i = \frac{-\partial H}{\partial q_i}(q_1,\ldots, q_n, p_1,\ldots, p_n), \tag{5}$$

$i = 1,\ldots, n$, or in vector notation,

$$\dot{q} = \frac{\partial H}{\partial p}(q,p)$$

$$\dot{p} = \frac{-\partial H}{\partial q}(q,p),$$

where the q_i denote generalized position coordinates, the p_i denote the generalized momentum coordinates and H represents the total energy of a system. (Using appropriate analogies, lossless electrical systems, electromechanical systems, etc., can be represented by (5) as well). The motions of system (5) are always such that the energy is conserved, since

$$\frac{d}{dt}H(q,p) = \sum_{i=1}^{n} \left(\frac{\partial H}{\partial q_i}\dot{q}_i + \frac{\partial H}{\partial p_i}\dot{p}_i\right) =$$

$$= \sum_{i=1}^{n} \left(\frac{\partial H}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial q_i}\right) = 0.$$

A linear Hamiltonian differential equation is obtained from a quadratic Hamiltonian of the form

$$H(p,q) = \frac{1}{2}q^T H_1 q + \frac{1}{2}p^T H_2 p \tag{6}$$

where we can assume, without loss of generality, that the matrices H_1 and H_2 are symmetric. Since in general $\frac{dH}{dt} \equiv 0$, we can use H as a Lyapunov function for (5) (when the potential energy has a local isolated minimum at the equilibrium $(q^T,p^T) = (0^T,0^T)$.) In particular, in the linear case (6) we ensure stability of the trivial solution with the assumptions that H_1 and H_2 are positive definite.

The linear system of differential equations corresponding to the Hamiltonian (6) is given by

$$\dot{q} = H_1 p$$

$$\dot{p} = -H_2 q. \tag{7}$$

Now if viscous damping is added, then system (7) will be replaced by

$$\dot{q} = H_1 p$$

$$\dot{p} = -H_2 q + Kp, \tag{8}$$

where $B_1 = H_2K + K^TH_2$ is negative semidefinite. In this case, the derivative of $H(q,p)$ along the solutions of (8) is given by

$$D_{(8)}H = p^TB_1p \leq 0.$$

If in particular we specialize (6) to a simple mechanical system consisting of n rigid bodies with masses m_i, $i = 1,\ldots, n$, then (6) will assume the form

$$H(q,p) = \frac{1}{2}q^THq + \frac{1}{2}p^TM^{-1}p, \tag{9}$$

where $M = \mathrm{diag}\ [m_1, m_2,\ldots, m_n]$ and $H = H^T$ (which characterizes the potential energy term) is determined by the system configuration. For this case Eq. (8) assumes the form

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= -Hq + KM^{-1}p \end{aligned} \tag{10}$$

which is equivalent to Newton's Law ($p = Mq$, $K = K^T$),

$$M\ddot{q} + Hq - K\dot{q} = 0.$$

The derivative of (9) along the solutions of (10) is easily computed to be

$$D_{(10)}H = p^TM^{-1}KM^{-1}p.$$

We now state and prove our final result.

Theorem 3. Consider the Hamiltonian (9) and the system (10) with $M = \mathrm{diag}[m_1, m_2,\ldots m_n] > 0$, $H = H^T > 0$, $K = K^T \leq 0$. Then the trivial solution of (10) is asymptotically stable if and only if the pair $(K, M^{-1}H)$ is observable.

Proof. Applying Theorem 2 with

$$A = \begin{bmatrix} 0 & M^{-1} \\ -H & KM^{-1} \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & M^{-1}KM^{-1} \end{bmatrix}, \qquad C = D = \begin{bmatrix} 0 & 0 \\ 0 & KM^{-1} \end{bmatrix}$$

it is easy to see that for $j = 0,1,2,3,\ldots$, we have

$$C(A-D)^{2j+1} = \begin{bmatrix} 0 & 0 \\ K(-M^{-1}H)^{j+1} & 0 \end{bmatrix}$$

and

$$C(A-D)^{2j} = \begin{bmatrix} 0 & 0 \\ 0 & K(-M^{-1}H)^jM^{-1} \end{bmatrix}.$$

Thus, the result follows from Theorem 2. ■

On the basis of this theorem, we can formulate the following simple rule for conservative stable systems of the form (9):

Pick a position in the undamped system where it is possible to add damping (e.g., dashpot, resistor, etc.). If it is always possible to detect motion at this position whenever the system is not at rest, then this is a location at which damping, to stabilize the system, can be added. To cover multi-position cases, the above must be modified, using linear combinations of motions at allowable damping points.

The above rule is easily seen to work for the example discussed in the introduction. One can also check the algebra for this example to obtain the results precisely. Indeed, we have

$$K = \begin{bmatrix} (-B_1-B) & B \\ B & (-B_2-B) \end{bmatrix},$$

$$M^{-1}H = \begin{bmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{m_2} \end{bmatrix} \begin{bmatrix} (k_1+k) & -k \\ -k & (k_2+k) \end{bmatrix},$$

and we consider the following possibilities:

Case 1: det $K \neq 0$

Case 2a: det $K = 0$ with $B_1 = B_2 = 0$

Case 2b: det $K = 0$ with $B_1 = B = 0$

Case 2c: det $K = 0$ with $B_2 = B = 0$.

For case 1, the pair $(K, M^{-1}H)$ is observable.

For case 2a, we have

$$K(M^{-1}H) = B \begin{bmatrix} \frac{-(k_1+k)}{m_1} - \frac{k}{m_2} & \frac{(k_2+k)}{m_2} + \frac{k}{m_1} \\ \frac{(k_1+k)}{m_1} + \frac{k}{m_2} & \frac{-(k_2+k)}{m_2} - \frac{k}{m_1} \end{bmatrix}.$$

In order that the pair $(K, M^{-1}H)$ not be observable, it is necessary and sufficient that the first row of $KM^{-1}H$ be negative of the second row, i.e.,

$$\frac{k_1+k}{m_1} + \frac{k}{m_2} = \frac{k_2+k}{m_2} + \frac{k}{m_1},$$

or

$$\frac{k_1}{m_1} = \frac{k_2}{m_2}.$$

This is the condition under which it is possible for the two masses to move in synchronism.

For case 2b it is easy to compute that the conditon which ensures that the pair $(K,M^{-1}H)$ be observable is $B_2k > 0$.

For case 2c the condition which ensures that the pair $(K,M^{-1}H)$ be observable is $B_1k > 0$.

Applying Theorem 3 we see that the system of Fig. 2 will be asymptotically stable for all the above cases, except case 2a.

IV. CONCLUDING REMARKS

We point out that in the present results, the asymptotic stability of the equilibrium of a system is ascertained by (a) determining the stability of the equilibrium of the corresponding conservative system, and (b) determining the attractivity of the equilibrium by examining the topological properties of the entire damped system. We hope to be able to extend these results to nonlinear systems as well. Such results would make it possible to investigate the stability of a large and important class of dynamical systems routinely, circumventing completely the usual Lyapunov approach. In addition, we believe that even for the case of linear systems, results of the type considered here provide useful insight into the qualitative behavior of such systems.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grant ENG 77-28446 and by the Engineering Research Institute, Iowa State University, Ames, Iowa 50011.

REFERENCES

[1] E. A. Barbasin and N. N. Krasovskii, "Stability of Motion in the Large," DAN 86 (1952), p.3.

[2] H. Goldstein, Classical Mechanics, Addison-Wesley, Reading, Mass., 1950.

[3] W. Hahn, Stability of Motion, Springer-Verlag, New York, 1967.

[4] J. P. LaSalle, "Asymptotic Stability Criterion," Proc. Symposia in Applied Mathematics, Vol. 13, American Math. Soc., Providence, R.I., 1962, pp. 299-307.

[5] J. J. Levin, "On the Global Asymptotic Behavior of Nonlinear Systems of Differential Equations," Arch. Rat. Mech. Anal. 6 (1960), pp. 65-74.

[6] D. Russell, unpublished communication to the authors.

[7] E. T. Whittaker, A Treatise on Analytical Dynamics of Particles and Rigid Bodies, Cambridge Univ. Press, 4th Edition, 1970.

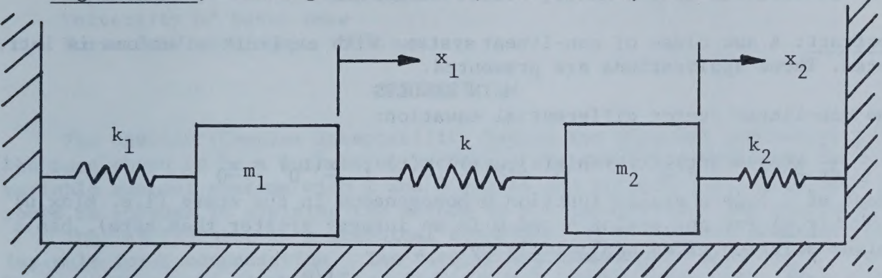

x_1
x_2
k_1
m_1
k
m_2
k_2

Figure 1

$B_1 \geq 0, \ B \geq 0, \ B_2 \geq 0, \ B_1 + B + B_2 > 0$

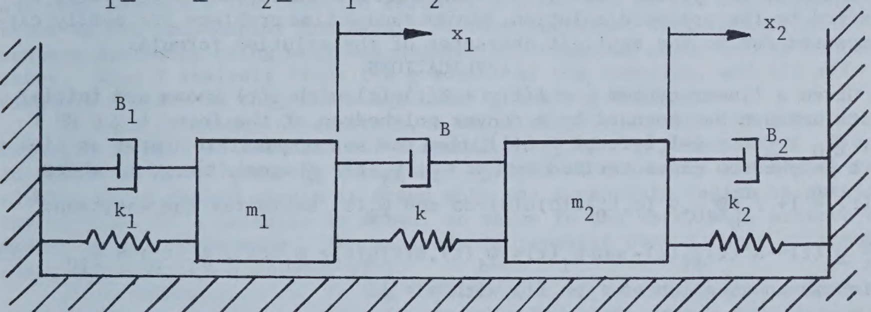

x_1
x_2
B_1
B
B_2
k_1
m_1
k
m_2
k_2

Figure 2

ON A NEW CLASS OF SEMI-LINEAR SYSTEMS AND THEIR APPLICATIONS

HEBERTT SIRA-RAMIREZ
Universidad de Los Andes, Mérida-Venezuela

Abstract: A new class of non-linear systems with explicit solutions is introduced. Three applications are presented.

MAIN RESULTS

The non-linear vector differential equation:

$$\frac{d}{dt}\underline{x}(t) = A(t)\underline{x}(t) + h(\underline{x}(t),\underline{u}(t))\,\underline{x}(t) \;;\quad \underline{x}(t_0) = \underline{x}_0 \tag{1}$$

where h(,) is a scalar function m-homogeneous in the state (i.e. $h(\alpha\underline{x},\underline{u}) = \alpha^m h(\underline{x},\underline{u})$ for any scalar α and m is an integer greater than zero), has a unique solution given explicitely by:

$$\underline{x}(t) = \left(1 - m\int_{t_0}^{t} h(\,\Phi(\sigma,t_0)\underline{x}_0,\ \underline{u}(\sigma))\,d\sigma\,\right)^{-1/m} \Phi(t,t_0)\underline{x}_0 \tag{2}$$

where $\Phi(t,t_0)$ is the state transition matrix associated with A(t). The proof of this fact is easily established through standard Lipschitz conditions type of arguments.[1] A bounded gradient of h over any bounded region of the state space suffices for the r.h.s. of (1) to be Lipschitzian. Straightforward derivation of (2) yields back (1) and the initial conditions are trivially satisfied by the proposed solution. Finite escape time problems are easily circumvented due to the explicit character of the solution formula.

APPLICATIONS

1) Given a linear system $\dot{\underline{x}} = A(t)\underline{x} + B(t)\underline{u}(t)$ with $\underline{u}(t)$ known and initial state unknown but bounded by a convex polyhedron of the form $\{\ \underline{x} \in R^n : \langle\underline{x},\ \underline{\psi}_{i0}\rangle \leqq 1\ ;\ i=1,2,\dots,N \geqq n+1\}$, then the set of possible states at time t is a polyhedron characterized by $\{\ \underline{x} : \langle\underline{x},\ \underline{\psi}_i(t)\rangle \leqq \operatorname{sgn} m_i(t)\ ;\ \forall\ i\}$ where $m_i(t) \overset{\Delta}{=} 1 + \int_{t_0}^{t}\langle\underline{\psi}_{i0}, \Phi'(\sigma,t_0)B(\sigma)\underline{u}(\sigma)\rangle d\sigma$ and $\underline{\psi}_i(t)$ satisfies the equation:[2]

$$\frac{d}{dt}\underline{\psi}_i(t) = -A'(t)\underline{\psi}_i(t) - \operatorname{sgn} m_i(t)\langle\underline{\psi}_i(t), B(t)\underline{u}(t)\rangle\,\underline{\psi}_i(t)\ ;\quad \underline{\psi}_i(t_0) = \underline{\psi}_{i0} \tag{3}$$

which is an equation of type (1) with m = 1.

2) State constrained (hemispace restriction) optimal control problems defined on linear systems with initial state restricted to a bounded region of a linear variety characterized by a support vector $\underline{h}_0$ are easily shown to yield differential equations of the form:

$$\frac{d}{dt}\underline{h}(t) = -A'(t)\underline{h}(t) - \langle\underline{p}(t),\underline{h}(t)\rangle\langle\underline{h}(t),B(t)R^{-1}(t)\underline{h}(t)\rangle\underline{h}(t)\ ;\quad \underline{h}(t_0) = \underline{h}_0 \tag{4}$$

where $\underline{h}(t)$ is the support vector of the state variety at time t, $\underline{p}(t)$ is a co-state vector arising from the optimization problem defined on the basis of a performance index that penalizes 1) any deviation of $\underline{h}(t)$ from a vector $\underline{q}(t)$ defining the hemispace restriction for the state trajectories and 2) the control power by means of a generalized norm defined by $R(t)=R'(t) > 0$. Equation (4) is of form (1) with m=3.

3) The main results are easily extended to matrix differential systems[3] of the form: $d/dt\ \Sigma(t) = A(t)\Sigma(t) + \Sigma(t)A'(t) + h(\Sigma(t),U(t))\Sigma(t)$ with $\Sigma(t_0) = \Sigma_0$.

REFERENCES

(1) Nemytskii, V., and Stepanov, V., Qualitative Theory of Diff. Eq. Princeton Press. 1960.

(2) Sira-Ramirez, H., "Reachable Sets and Set-Theoretc Evolution of the Uncertainty in Linear Systems," II Brazilian Congress of Automatic Control Universidade Federal de Santa Catarina (UFSC), Florianópolis. Sept. 1978

(3) Athans, M., "The Matrix Minimum Principle," Information and Control. Vol. 11 Numbers 5 & 6. November-December 1967.

CARDIAD DESIGN: PROGRESS IN THE FOUR INPUT/OUTPUT CASE*

R.M. SCHAFER AND M.K. SAIN
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

The CARDIAD (Complex Acceptability Region for DIAgonal Dominance) plot has been shown to be a successful technique for achieving dominance in multivariable control systems with 2 and 3 inputs and outputs [1]. This note reports on success in applying the method to a 4 input-output jet engine model.

The CARDIAD plot is a graphical technique which achieves dominance using only input compensation. The form of the compensator used has 1's on the main diagonal and general complex entries off the diagonal. At a given frequency, a sufficient condition for dominance in a column of the system can be expressed as a quadratic function of the system evaluated at the frequency with the off-diagonal entries of the compensator as variables. To achieve dominance, the function must be made positive.

Two types of analysis are performed on the dominance function. Type 1 analysis maximizes the function assuming all but one of the off-diagonal entries in the compensator are zero. In this approach, it is attempted to achieve dominance using only one nonzero off-diagonal entry in the compensator. Type 2 analysis finds the gradient of the function, and all off-diagonal entries in the column are used to achieve dominance.

At each frequency of analysis, a solid circle, a dashed circle, or a triangle is plotted for each off-diagonal entry of the compensator. A solid circle is drawn when the acceptable choices of the entry lie inside the circle, and a dashed circle is drawn when the acceptable region is outside the circle. If a triangle is drawn, no value in the entry will achieve dominance at that frequency. The analysis is repeated over a range of frequencies, resulting in a CARDIAD plot.

If no triangles occur in the CARDIAD plot of an entry, that column of the system can be made dominant by choosing for that entry a function which lies inside the solid circles or outside the dashed circles at each frequency. If there exists a point on the real axis which is inside all solid circles and outside all dashed circles, constant compensation will achieve dominance. If no such point exists, the plot indicates the proper choice of the function by describing its path through the complex plane. In the case where all off-diagonal entries contain triangles, and hence no one entry is able to achieve dominance, all entries must be designed to fit the shape of the path through the complex plane described by the triangles in the type 2 analysis plots.

CARDIAD is a graphical technique which preserves much of the insight in the design procedure. The plots readily predict whether constant compensation can be used, what values work if it can, and what the frequency dependent compensation should look like if constant compensation won't work. Typical of this case is a plot which has solid circles forming a semi-circular plot in the complex plane. A function of the form $k_{ij}(s)=\{\alpha+\beta s\}/\{1+\delta s\}$ can be chosen by fitting the function at points on the plot.

*This work was supported by NASA Grant NSG 3048.

1. R.M. Schafer and M.K. Sain, "Input Compensation for Dominance of Turbofan Models", in Alternatives for Linear Multivariable Control, National Engineering Consortium, 1978, pp. 156-169.

A MODEL FOR THE PARALLEL EXECUTION OF PROGRAMS WITH ASYNCHRONOUS OPERATIONS*

V. VAIRAVAN†
Johnson Controls,
Milwaukee, Wisconsin

K. VAIRAVAN
Department of Electrical Engineering and
Computer Science
University of Wisconsin-Milwaukee

ABSTRACT

The parallel execution of programs has received much attention recently. It has been shown that the generation of a complete optimal parallel schedule of a program is of exponential complexity even in a restricted model of computation [1]. In view of the complexity of constructing a complete parallel schedule, and since the number of available processors may not be known in advance or may not be fixed, there is an interest in intermediate parallel representations of programs in which information about concurrency is included but not all the scheduling details. In this paper we present efficient techniques to obtain such an intermediate level representation called a parallel execution graph. We show that this representation can be easily conversed into a table suitable for dynamic interpretation. Only an extended summary is presented here. The details can be found in [3].

INTRODUCTION

In recent years there have been numerous studies of various aspects of parallel computation. The parallel execution sequential programs has been studied from an empirical point of view by several authors including Ramamoorthy and Kuck. The same topic has been studied along more theoretical lines by DeMillo and Vairavan among others. DeMillo and Vairavan have developed a framework for studying the scheduling problem in a synchronous model of computation [1,2]. Their work lies between the heuristic approaches of Ramamoorthy on the one hand and the vastly more extensive theoretical studies of parallel schema by Karp and Miller. DeMillo and Vairavan have shown that optimal (minimal execution-time) schedules may not exist for cetain programs on some fixed numbers of processors and even if such optimal schedules exist, they may be of exponential complexity. In view of the exponential complexity of optimal parallel scheduling and because the number of available processors may not be fixed, or may not be known in advance, there is a great deal of interest in developing intermediate level parallel representations of programs which include explicitly some information about concurrency but not all the scheduling details. The inter-

---

* Supported by NSF grant no. MCS 78-28305

† Formerly with the Department of Electrical Engineering and Computer Science, University of Wisconsin-Milwaukee.

mediate representation can then be dynamically interpreted and scheduled by an operating system in a multiprocessor environment. This approach will also permit the consideration of programs with operations whose execution times are not necessarily equal and may not be known in advance.

Our paper will present such an intermediate parallel representation of programs. Procedures will be presented which will automatically transform sequential flow charts into a parallel representation called the parallel execution graph. In the first procedure P_1, the focus will be on the control structure of the flow charts. The second procedure P_2 will deal with data dependencies between operations in the flow chart and their transformation into the parallel representation. The procedure P_3 will deal with a technique called "path balancing" which is necessary to handle unequal numbers of data dependencies along different control paths in a flow chart. Applications of P_1, P_2 and P_3, in that order, would yield a parallel execution graph which, we indicate, can be converted into a parallel execution table. Because of space limitations we do not present details and some other related ideas here, but which are dealt with in (3).

CONTROL-STRUCTURE MAPPING

By a program graph we mean a flow chart with uninterpreted operations. The flow chart is characterized by a unique starting node S and a inique terminal node F. A careful definition can be found in (1,2). In contrast to the references cited, in this paper we permit the operations to be asynchronous. Although the details of the operations are not assumed to be known, data dependencies between the operations are assumed to be known and expressed through a binary relation ρ on the operation set. In the illustrations that follow, a data dependency between two operations is indicated by a directed broken line. (See Figure 1.) Given a program graph G, procedure P_1 transforms G into a graph T in which the control structure is represented more conveniently for parallel execution. Before stating P_1, we present some informal definitions. A node y is said to be strongly reachable (SR) from a node x in a flow chart G if every path from x to the terminal node F contains y. If x is a decision node and α is an outcome of x and a label on an edge (x,z) in G, then a node y is strongly reachable from x only along α if i) y is SR from z and ii) y is not S.R. from x. Finally a dummy node in T would indicate a null operation. A node x is said to be duplicated in T if another copy of x exists in T.

PRODECURE P_1

1. Link to S, nodes that are SR from S.
2. Let D be an "ordered" list of decision nodes linked to S in step 1.

```
3.  While $D \neq \phi$ do
    1.  For each decision node d in D and for each outcome $\alpha$ do
        (i)  If d is duplicated
             Then set up an arc labelled $\alpha$ from d to a corresponding
                  unique dummy node successor $\delta^{\alpha}$ of a duplicate of d
             Else
                  (1) Set up a dummy node $\delta^{\alpha}$ and link $\delta^{\alpha}$ to d with
                      the label $\alpha$ and
                  (2) If outcome $\alpha$ is in a cycle $\zeta$
                      Then link to $\delta^{\alpha}$ all nodes SR from d only along
                           $\alpha$ and which are in cycle $\zeta$ including possibly d;
                      Else link nodes to $\delta^{\alpha}$ such that these nodes are
                           SR from d only along $\alpha$.
        (ii)  Delete from D this appearance of d
    II. Add to D every decision node resulting from I.
```

It can be easily shown that P_1 always terminates. The procedure of P_1 results in a graph T. If in T a node x appears ℓ times, then the node appearances are arbitrarily labelled $x_1, x_2, \ldots x_{\ell}$.

MAPPING DATA-DEPENDENCIES

Data dependencies from G are mapped to the transformed graph T using procedure P_2. Some terms used in P_2 are informally defined before stating the procedure. A graph w is a loop in a graph G if

(i) w is a subgraph of G and

(ii) w is strongly connected.

v is an entry node of a loop w if $\exists$ a node v_x in G and not in w, such that v is an immediate successor of v_x. v is an exit node of a loop w if $\exists$ a node v_y in G, but not in w, such that v_y is an immediate successor of v. Let w and w' be loops in G, w' is larger than w if w is a proper subgraph of w'. An operation z is said to be linked to x in T if z is reachable from x in T through a series of non-dummy nodes. We say that a node y can follow execution of x in T (written as $x \xrightarrow[T]{*} y$) if (1) $\exists$ a path from x to y in T or (2) $\exists$ a decision d in T such that x is reachable from d and y is linked to d.

PROCEDURE P_2

```
1.  Let $(x,y) \in \rho$.
    If x is in some largest loop w in G and y is not in w
        Then for each exit node from w, set up an arc from the corres-
        ponding exit node $\delta^{e}$ in T to every copy $y^{j}$ of y such that
```

$\delta^e \xrightarrow[T]{*} y^j$ and from each copy x^e of x that is not in a loop in T, set up an arc from x^e to y^j where $x^e \xrightarrow[T]{*} y^j$

Else from each copy X^e set up an arc from x^e to y^j where $x^e \xrightarrow[T]{*} y^j$.

2. If x^i and x^j are copies of the same non decision operation x and x^j is reachable from x^i in T, then set up a dependency arc from x^i to x^j.

PATHBALANCING

Next we consider with a procedure P_3 for Path Balancing which deals with unequal number of data dependencies along different control paths in a program. A recursive definition of dependency indices (ψ^x_y) is given first. These indices will be useful in path balancing and later in setting up a parallel execution table.

(1) Initialize $\psi^x_y = 0$ for every pair x,y in T.

(2) For each pair of nodes (q,r) in T such that $q \xrightarrow[T]{*} r$ define ψ^q_r as follows:

  (i) q - a non-decision node.

  If there exists an immediate link t ($\neq$r) to the node q then define $z = \psi^t_r$; otherwise z = 0; $\psi^q_r = z+1$ if $\exists$ a dependency arc from q to r in T; otherwise $\psi^q_r = z$.

  (ii) q - a decision node not lying in a cycle in T

$$\psi^q_r = \mathrm{Max}\left(\psi^{\Delta_1}_r, \psi^{\Delta_2}_r\right) + z.$$

  z is defined as in (i) and Δ_1 and Δ_2 are the dummy nodes associated with the decision q.

  (iii) q - a decision node which is in a cycle w in T and r is not in w.

$$\psi^q_r = \mathrm{Max}\left[\psi^{\delta_j}_r\right]$$

  where δ^j denotes a dummy node associated with an exit from the largest loop in T in which q appears but not r.

  (iv) q (a decision) and n are in some same cycle w in T.

  Let z denote an immediate successor of q in w is T.

  Then $\psi^q_n = \psi^z_n + \max_{\delta^j}\left[\psi^{\delta^j}_r\right]$ where δ^j denotes a dummy node associated with an exit from the largest loop in T that contains q and r.

PROCEDURE P_3

1. "Mark" every decision node that is in a cycle in T.
2. Consider a decision d (i) which is not in a cycle and (ii) such that every decision reachable from d in T is marked:

  (A) Consider each P in T such that $\delta^j \xrightarrow[T]{*} P$ (j = 1,2) where δ^1,δ^2 are the dummy nodes associated with d, ($P \neq \delta^j$). Define

$y^j = \psi_P^{\delta^j}$ (j=1,2). If $y^i = y^j$, no balancing is needed. Go to B. If $y^i > y^j$, define $\Delta = y^i - y^j$. Introduce a dependency arc from δ^j to P with weight Δ. Go to B.

(B) When all such node P are processed, mark d and go to 2 until all decision nodes are processed.

See figures 1 and 2 for an illustration of the ideas discussed above. Figures 1b and 2b are the corresponding parallel execution graphs that result from the applications of procedures P_1, P_2 and P_3 to Fig. 1a and 2a.

PARALLEL EXECUTION TABLE

The parallel execution graph of a program can be converted into a parallel execution table (PET) which is suitable for dynamic interpretation. For each node x in the parallel execution graph there is a row labelled x in the table with several entries as discussed below:

Control Link: This will be a null entry unless x is a decision node. If x is a decision, the entries point to possible successor nodes of x. The actual successor node will depend upon the outcome of x and will be determined only upon the completion of x.

Precedence Count: This is a counter K_x indicating the number of data dependencies to be satisfied before the execution of x can begin. This counter K_x is decremented whenever a data dependency affecting x is satisfied and when K_x becomes zero, x becomes eligible for execution. The initial value of $K_x = \psi_x^S$ where S is the starting node.

Scheduling Link: This entry points to a linked list of operations that are "controlwise" eligible for execution. This list would be a list of nodes strongly reachable from x. These operations would be placed in a scheduling list (queue) of operations to be executed.

Dependencies Satisfied: Here we have a list of entries of the form (y,n). The significance of (y,n) is that upon the completion of x, the precedence counter of y is decremented by n. The initial entries are obtained from the parallel execution graph. If n is a negative integer, then the effect is to increment the dependency counter - a step whose significance is explained in the following discussion.

REACTIVATION OF DATA DEPENDENCIES

If an operation x is in a cycle and is executed, it is possible that it may be executed again. Hence it is necessary to reactivate some of the data dependencies that affect x. This is done as follows: If x appears in a cycle w in the parallel execution graph T then let z be the immediate successor of x in w. Upon the execution of x, a "dependency satisfied" entry of $(x,-\psi_x^z)$ is included in row z. Figures 1c and 2c indicate the initial PETs for the parallel execution graphs shown in figures 1b and 2b respectively. Note that some of the table entries will be charged dynamically during program execution.

CONCLUSION

Procedures to convert a program graph into a representation more suited for parallel execution have been presented. Details of these procedures including their complexity and illustrations of parallel execution can be found in (3). Problems for further investigation include a study of execution models and overhead costs for dynamic interpretation of PET.

REFERENCES

1. Vairavan, K. and DeMillo, R. A. "On the computational complexity of a generalized scheduling problem", IEEE TC, C-25,11, Nov. 1976.
2. DeMillo, R. A., Vairavan, K., and Sycra-Cyranski, "A study of schedules as models of synchronous parallel computation", JACM, Vol. 24, No. 4, October, 1977.
3. Vairavan, K. and Vairavan, V., "An intermediate representation of programs for parallel execution", University of Wisconsin-Milwaukee, Dept. of EECS, Tech. Report No. 78-8.

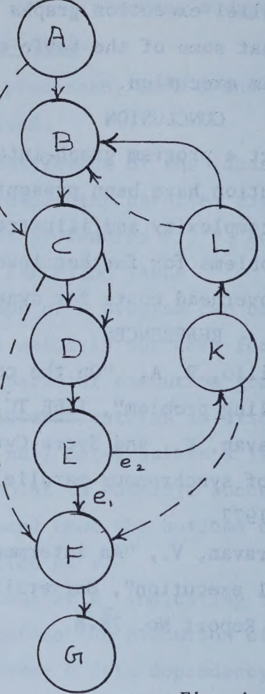

A
B
C
D
E
F
G
L
K
e₂
e₁

Fig. 1a: Flowchart M_1

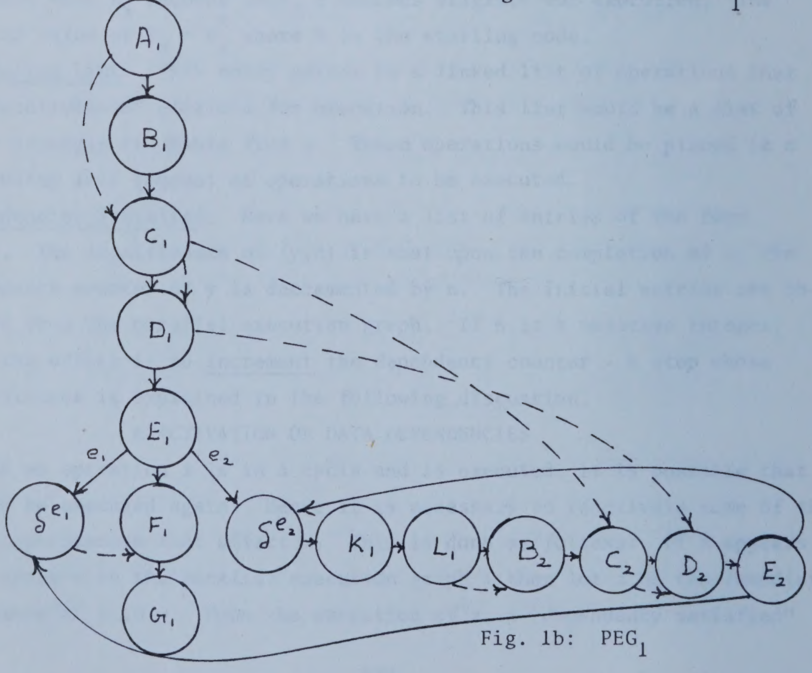

A₁
B₁
C₁
D₁
E₁
e₁
e₂
δ^e₁
F₁
G₁
δ^e₂
K₁
L₁
B₂
C₂
D₂
E₂

Fig. 1b: PEG_1

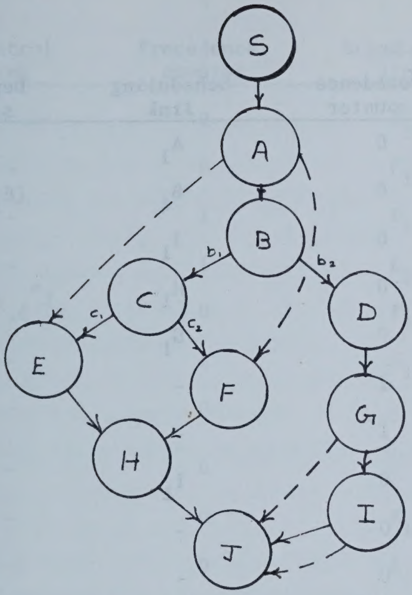

S
A
B
b₁
b₂
C
D
c₁
c₂
E
F
G
H
I
J

Fig. 2a: Flowchart M_2

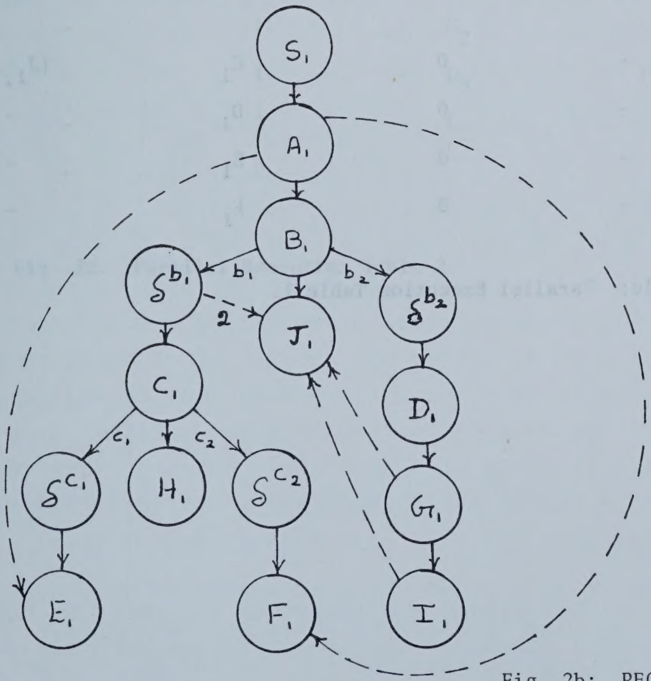

S₁
A₁
B₁
b₁
b₂
δ^b₁
2
J₁
δ^b₂
C₁
D₁
c₁
c₂
δ^c₁
H₁
δ^c₂
G₁
E₁
F₁
I₁

Fig. 2b: PEG_2

Node	Control link	Precedence counter	Scheduling link	Dependencies satisfied
S_1	-	0	A_1	-
A_1	-	0	B_1	$(E_1,1),(F_1,1)$
B_1	$\delta^{b_1},\delta^{b_2}$	0	J_1	-
C_1	$\delta^{c_1},\delta^{c_2}$	0	H_1	-
D_1	-	0	G_1	-
E_1	-	1	-	-
F_1	-	1	-	-
G_1	-	0	I_1	$(J_1,1)$
H_1	-	0	-	-
I_1	-	0	-	$(J_1,1)$
J_1	-	2	-	-
δ^{b_1}	-	0	C_1	$(J_1,2)$
δ^{b_2}	-	0	D_1	-
δ^{c_1}	-	0	E_1	-
δ^{c_2}	-	0	F_1	-

Fig. 1c: Parallel Execution Table 1.

Node	Control link	Precedence counter	Scheduling link	Dependencies satisfied
A_1	-	0	B_1	$(C_1, 1)$
B_1	-	0	C_1	-
C_1	-	1	D_1	$(C_2, 1), (D_1, 1)$
D_1	-	1	E_1	$(D_2, 1)$
E_1	$\delta^{e_1}, \delta^{e_2}$	0	F_1	-
F_1	-	1	G_1	-
G_1	-	0	-	-
δ^{e_1}	-	0	-	$(F_1, 1)$
δ^{e_2}	-	0	K_1	-
K_1	-	0	L_1	-
L_1	-	0	B_2	$(B_2, 1)$
B_2	-	1	C_2	-
C_2	-	1	D_2	$(D_2, 1)$
D_2	-	2	E_2	-
E_2	-	0	-	-

Fig. 2c. Parallel Execution Table 2.

STING-UNLESS: A CONDITIONAL, INTERLOCK-FREE STORE INSTRUCTION*

DANIEL P. FRIEDMAN and DAVID S. WISE
Computer Science Department
Indiana University

Abstract: We define *sting-unless* as an interlock-free, parallel store instruction which does not encumber processor/memory communication. Rather, store instructions are dispatched from many processors to many banks of memory. With a suitable processors/memories switch, this operation is realizable in *memory* hardware, so that programs may be designed and run without the control structures required by processor/processor or processor/memory interlocks. The paper includes a syntax, an operational semantics, an example, implementation of a binary semaphore, and a hardware circuit of a memory board suitable for the example.

CR categories: 4.32, 6.34.

We introduce a memory store instruction which derives from the memory *stinging* operation introduced elsewhere [4]. In that paper we introduced the idea of an independent processor probing a data structure only to find a suspension instead of the desired value, then evaluating the suspension and instantaneously displacing that suspension (or stinging the field containing it) with its intended value. The intention of the term *sting* there was to communicate the quickness of the memory instruction which proceeds without disturbance to other processors which might be near (but not at) the stung node, and the convention that a process necessarily die upon stinging after it had fulfilled its purpose in evaluating a suspension.

In this note we refine the concept of stinging to a hardware operation which is useful for surviving as well as dying processors. The name is derived from the original use, delivering results in place of suspensions; but its intended application is in a system where suspensions may be coaxed slightly rather than (necessarily) evaluated completely. In the new application [5] a reference to an advanced (coaxed) suspension may be stung over a predecessor, but the process might survive to coax the newer suspension again. Thus, repeated stinging may occur in the same pointer field until a terminal value is stung, whence all future stinging must stop. We retain the term *sting* to suggest the sharpness of each processor's store instruction--interleaved between other processors' use of *that* memory word--and the preventative effect of an ultimately stung value is much like an injection to vaccinate against further stings. We provide a syntax, a simple example, and a practical implementation of the sting operation in memory (not processor) hardware. The reader is referred to another paper [6] for a development of its utilization.

---

*Research reported herein was supported (in part) by the National Science Foundation under grants numbered MCS75-06678 A01 and MCS77-22325.

We introduce a multiprocessing control structure, *sting* which generalizes the memory *store* operation. We shall use PASCAL's [8] up-arrow convention, however, for *fetch*ing the contents of a record using a pointer to that record. If Q is a pointer to a node and NQ is a record of the type referenced by Q, then a fetch is denoted by

```
NQ := Q↑
```

The up-arrow is only used for fetches, perhaps better called a "yank" to suggest that (in a competitive multiprocessor environment) delivery of the value may be delayed some time after the request for the fetch. Where PASCAL would allow the corresponding memory store instruction

```
Q↑  := NQ
```

we would write

```
sting Q with NQ  .
```

The general form of the *sting* operation is

```
sting POINTER unless FLAG in FIELD with VALUE  .
```

POINTER must be a reference to a record with at least the FLAG and FIELD fields. FLAG must be of type Boolean, and FIELD must be the same type as VALUE. The information describing FLAG, FIELD, and VALUE are dispatched to the record in memory designated by POINTER, and the following steps occur *without interruption*:

```
if not POINTER↑.FLAG then POINTER↑.FIELD := VALUE .
```

The bit specified by FLAG in that record is tested. If it is *false* then VALUE is stored in FIELD there. If it is *true* then nothing happens. If the "*unless* FLAG" is omitted from the operation, then the field is stored unconditionally, like PASCAL's

```
POINTER↑.FIELD := VALUE .
```

If the "in FIELD" part is omitted then we presume that VALUE is an entire record and the whole record takes the new value:

```
if not POINTER↑.FLAG then POINTER↑ := VALUE .
```

Intuitively, POINTER is the address in memory of a word to be manipulated, FLAG is the bit number of a bit to be tested, POINTER↑.FIELD consists of a bit number of the leftmost bit of the field to be stored and the field-length of the number of bits to be changed, and VALUE is the value to be stored in the bit positions corresponding to the specified field.

With these memory access conventions it is easy to identify memory fetches (by the "↑") and memory stores (by *sting*); all other assignments (using " := ") are to local variables which we assume are high speed registers within a processor.

The operational semantics of *sting* is intended to be easily implementable in the memory hardware. The *unless* test is to be made remotely from any processor, possibly by "anding" the appropriate bit from a read-cycle back onto the write-enable line during the write-cycle (Figure 1). In this way each of many banks of memory may perform its own *sting* operation without interlocking the others or any processor. Processors need not wait after dispatching the (POINTER, FLAG, FIELD, VALUE) quadruple to memory, but they may proceed to interact with other banks of memory while their stings are processed. The reader should compare this implementation to one using processor-oriented controls (e.g. semaphores [3], *lock* and *unlock* [2], *test and set* [7], *compare and swap* [1], etc.) where only one *sting* may proceed at a time.

Sting is an indivisible test and store instruction which may be implemented remotely from any general processor. A processor which initiates a *sting* <u>never</u> finds out <u>directly</u> whether the value actually was stored. By precluding feedback on such operations where a processor is likely to encounter contention from other processors, we intend to free processors from interlocking and to allow them to proceed with productive computation.

Rather than interlocking a processor with a particular record in memory during a critical "write-enable" section of code, we specify that a write be attempted and then abandoned. Ordinarily such a write attempt will have as its destination a field in the node which includes the flag being tested, and the value being stored will include a *true* (*false*) value for that flag. In this way, if the store is permitted, it proceeds uninterruptibly but remotely from any processor, and future attempts of the same nature will be blocked (allowable). We attain a degree of protection against multiple successful stores without providing a semaphore on every record in memory and without delaying one processor while another seizes a record.

As an example of *sting* within a multiprocessing environment we shall model an Easter Egg Hunt. The hunt is a game with K children (processors) and MEMSIZE eggs (memory locations):

```
          type EGG = packed record
                              CLAIMED: Boolean;
                              OWNER:  1..K
                            end  .
```

The winner of the hunt is the child who claims the most eggs. In order to model the hunt we must give each child a move strategy. A good strategy is: if the J^{th} child has just claimed the I^{th} egg then the child moves to

```
          (I + PRIME[J]) mod MEMSIZE
```

where PRIME[J] is chosen from a set of K pairwise relatively prime numbers (one for each child) and MEMSIZE is relatively prime to each PRIME[J]. This strategy searches all eggs and should keep the memory contention small. It is a poor strategy to use the same strategy as any other child--one of two children with the same PRIME would probably be "scooped."

First the eggs are disbursed. That is, each cell in memory has the CLAIMED bit off. Then each child, J, is set loose:

```
begin
   I := PRIME[J];
   while I ≠ 0 do begin
                                 sting I unless CLAIMED with (true,J);
                                 I := (I + PRIME[J]) mod MEMSIZE
                              end
end   .
```

Finally, it is necessary to tally the scores and determine the winner. The pair (*true*,J) sets the CLAIMED bit and names the OWNER, J, if the memory location is yet unclaimed. The hunt proceeds on parallel processors which interact (by not stealing an egg claimed by another), but there are never any *processor* interlocks. The switch which allows communication with common memory is sufficient to assure a fair hunt.

A memory word, MUTEX, which has been declared as an EGG may also be used as a binary semaphore [3]. The P and V operations for the J^{th} processor may be implemented as

```
P(MUTEX)  ≡  repeat  sting MUTEX unless CLAIMED with (true, J)
                until MUTEX↑.OWNER = J ;
V(MUTEX)  ≡  sting MUTEX in CLAIMED with false .
```

We have defined *sting-unless* as an interlock-free, parallel store instruction which does not encumber processor/memory communication. Rather store instructions are dispatched from many processors to many banks of memory. With a suitable processors/memories switch, this operation is realizable in *memory* hardware, so that programs may be designed and run without the control structures required by processor/processor or processor/memory interlocks.

Acknowledgement:

We wish to thank Douglas Goode for Figure 1.

REFERENCES

1. Case, R. P., and Padegs, A. Architecture of the IBM Systems/370. *Comm. ACM 21*, 1 (January, 1978), 73-96.
2. Dennis, J. B., and Van Horn, E. C. Programming semantics for multiprogrammed computations. *Comm. ACM 9*, 3 (March, 1966), 143-155.
3. Dijkstra, E. W. Cooperating sequential processes. In F. Genuys (ed.), **Programming Languages**, Academic Press, New York (1968), 43-112.
4. Friedman, D. P. and Wise, D. S. Aspects of applicative programming for parallel processing. *IEEE Trans. Comput. C-27*, 4 (April, 1978), 289-296. Also in P. Enslow (ed.) **Proc. 1976 Intl. Conf. on Parallel Processing**, 263-272.
5. Friedman, D. P. and Wise, D. S. Applicative multiprogramming. Tech. Report 72, Computer Science Dept., Indiana University (January, 1978).
6. Friedman, D. P. and Wise, D. S. Fair applicative multiprogramming (in preparation).
7. Shaw, A. C. **The Logical Design of Operating Systems**, Prentice-Hall, Inc., Englewood Cliffs, NJ (1974), 80-82.
8. Wirth, N. The programming language PASCAL. *Acta Informatica 1*, 1 (1971), 35-63.

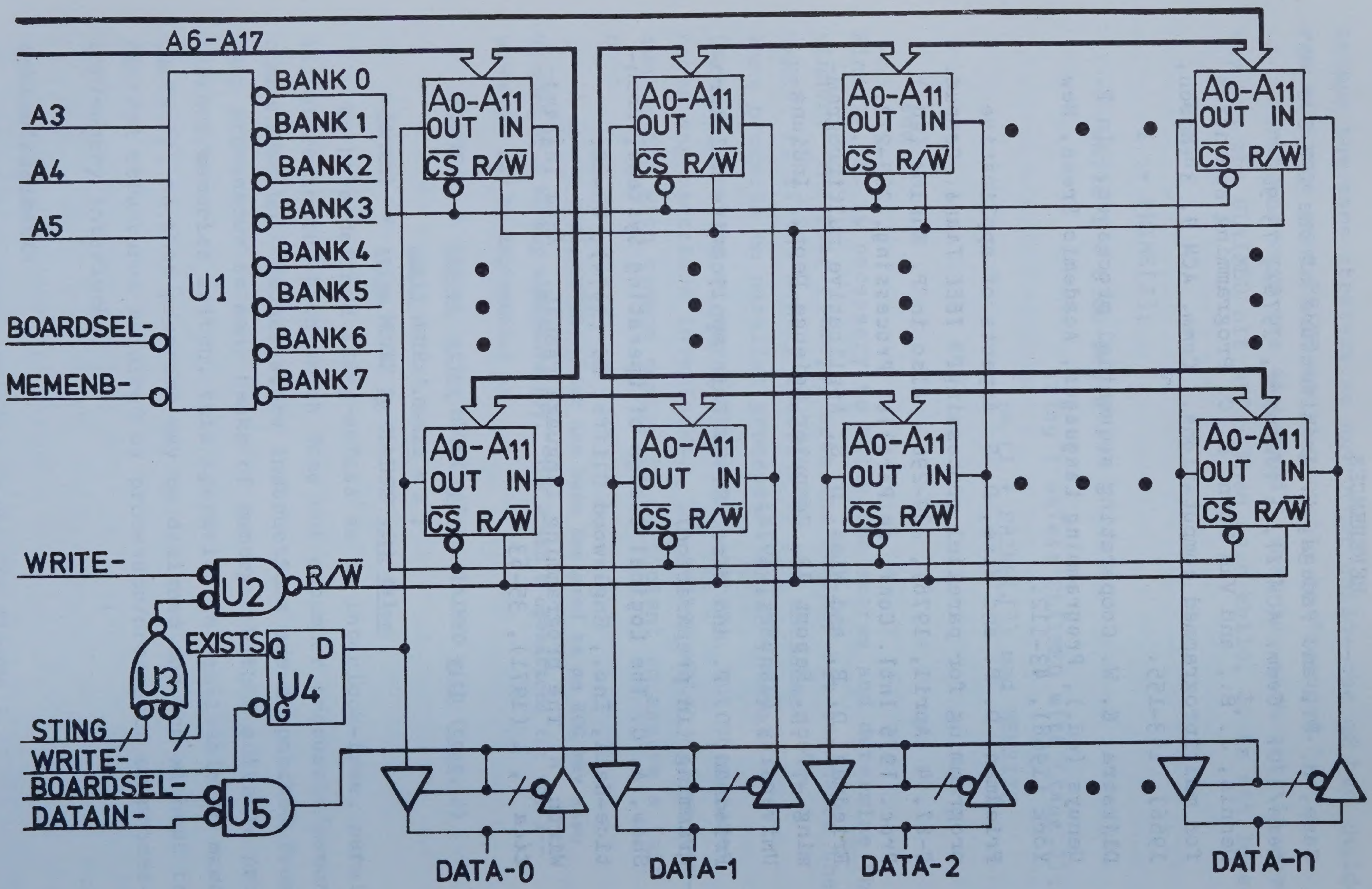

A6-A17
A3
A4
A5
BOARDSEL-
MEMENB-
U1
BANK 0
BANK 1
BANK 2
BANK 3
BANK 4
BANK 5
BANK 6
BANK 7
A0-A11
OUT
IN
CS
R/W
WRITE-
U2
R/W
U3
EXISTS
Q
D
U4
G
STING
WRITE-
BOARDSEL-
DATAIN-
U5
DATA-0
DATA-1
DATA-2
DATA-n

Figure 1. A memory for *sting-unless*

PARALLEL SORTING

LEE D. CORAOR
Electrical Sciences and Systems Engineering
Southern Illinois University at Carbondale
Carbondale, Illinois 62901

JOHN P. ROBINSON
Information Engineering
University of Iowa
Iowa City, Iowa 52242

SUMMARY

A multiple processor hardware sort organization is presented which reduces the total sort time and minimizes data movement time when compared to a single processor sort routine. The primary gain in speed is due to the availability of a fast search procedure, interval n-ing, which can quickly search both sparse and compact lists [1,2]. Memory is assumed to be inexpensive such that 100% memory utilization is not necessary.

The sort procedure uses an insertion sort technique in conjunction with parallel multiple processors and an interval n-ing search procedure. The interval n-ing search procedure can search a list of M items in at most P passes with n processors where

$$P = \left\lceil \mathrm{Log}_2(M+1)/\mathrm{Log}_2(n+1) \right\rceil$$

The number of data moves performed is reduced by the use of multicopying of data onto the list.

A simple example illustrates that using an organization with 4 processors and a memory with 64 locations, a random list of 52 items can be sorted with only 23 parallel data moves (80% memory capacity). For a random list of 64 items only 85 parallel data moves are needed (100% memory capacity). In comparison using a merge sort routine with the same data, 252 and 315 data moves are necessary for 52 and 64 list items respectively.

[1] Coraor, L. D., and J. P. Robinson, "List Searches Using Multiple Processors," Proceedings Allerton Conference on Communication, Control, and Computing, p. 162, 1977.

[2] Coraor, L. D., "A Multiprocessor Organization for Large Data List Searches," Ph. D. Dissertation, Electrical Engineering, University of Iowa, Iowa City, Iowa 52242.

PARTITIONABLE SIMD COMPUTER SYSTEM INTERCONNECTION NETWORK UNIVERSALITY

HOWARD JAY SIEGEL
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Single and multiple control partitionable SIMD computer systems are defined. Algorithms are presented to demonstrate how a variety of SIMD machine interconnection networks can perform an arbitrary interconnection in $O(r^2)$ time, where 2^r is the number of processors in the partition of the system. Five types of networks which have been discussed in the literature are analyzed and compared.

I. INTRODUCTION

The advent of the microprocessor is leading computer technology to the day when it will be feasible to construct parallel computer systems consisting of 2^{10} to 2^{14} processing elements [15]. The SIMD (single instruction stream-multiple data stream) machine [7] structure presents one way to harness this parallel processing power. A control unit broadcasts an instruction to the processing elements (PEs) and all active PEs execute the same instruction simultaneously, but each PE executes it on data in its own memory.

Whether or not all N PEs of an SIMD machine are required for a given computation is highly task dependent. If only 2^r of the $N(=2^m)$ PEs are needed, the other $N-2^r$ may be used to perform other tasks. Single control partitioning creates one to 2^{m-r} SIMD machines, each of size 2^r, all performing the same program, but on different data sets. Under multiple control partitioning a system can be partitioned into many SIMD machines of different sizes, each with its own control unit (hence, its own instruction stream) [22,23].

The interconnection network of a partitionable SIMD system should provide fast interprocessor communications without "unreasonable" cost. The set of interconnections which is implemented is used to perform or "simulate" other interconnections, if and when they are needed. The universality of an interconnection network is its ability to simulate arbitrary one-to-one interconnections [19].

This paper extends the work in [13,14,19,26,27]. In [16,17] the capability of an interconnection network to simulate a particular interconnection was discussed. In this paper, algorithms which permit various networks to simulate arbitrary interconnections are presented. The universality of five types of interconnection networks in a partitionable SIMD computer system is analyzed.

This work was supported by the Purdue Research Foundation under an XL Grant and by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-78-3581. The United States Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation hereon.

II. SIMD MACHINES

The model of SIMD machines used as a framework for the algorithms presented in this paper was introduced in [17] and is briefly described here. The model consists of four components: processing elements, interconnection functions, machine instructions, and masking schemes.

There are $N = 2^m$ *processing elements* (*PEs*), numbered (addressed) from 0 to N-1. Each PE consists of a processor, memory, fast access registers (*A*, *B*, *C*), a data transfer register (*DTR*), and a register containing the PE's address (*ADDR*). ADDR(i) denotes the i-th bit of ADDR. If a PE is *active* it executes the instructions broadcast to it, if it is *inactive* it does not.

An *interconnection network* is a set of *interconnection functions*, each a bijection on the set of PE addresses. When an interconnection function f is executed PE i copies the contents of its DTR into the DTR of PE f(i), for all i, $0 \leq i < N$ and PE i active, simultaneously. An inactive PE may receive data from an active PE, overwriting the inactive PE's DTR contents. Five interconnection networks will be defined, using $p_{m-1} \cdots p_1 p_0$ as the binary representation of an arbitrary PE address and $\overline{p}_i$ as the complement of p_i.

The *Cube* network consists of m functions defined by:

$$\mathrm{cube}_i(p_{m-1} \cdots p_1 \ p_0) = p_{m-1} \cdots p_{i+1} \overline{p}_i p_{i-1} \cdots p_0,$$

for $0 \leq i < m$. The STARAN network [3] and the indirect binary n-cube network [15] are multistage implementations of the Cube. The usefulness of this type of network is discussed in [2,3,15-17,25].

The *Plus-Minus* 2^i (*PM2I*) network consists of 2m-1 functions defined by:

$$\mathrm{PM2}_{+i}(j) = j+2^i \bmod N, \qquad \mathrm{PM2}_{-i}(j) = j-2^i \bmod N,$$

for $0 \leq i < m$, $0 \leq j < N$. (Note $\mathrm{PM2}_{+(m-1)} = \mathrm{PM2}_{-(m-1)}$.) The data manipulator network [6] is a multistage implementation of the PM2I. Networks similar in concept are included in the Omen and SIMDA machines [3,28]. The usefulness of this type of network is discussed in [6,16,17,24,25].

The *Wrap-Around Plus-Minus* 2^i (*WPM2I*) network consists of 2m functions defined by:

$$\mathrm{WPM2}_{+i}(p_{m-1} \cdots p_1 p_0) = q_{m-1} \cdots q_1 q_0,$$

$$\text{where } q_{i-1} \cdots q_0 q_{m-1} \cdots q_{i+1} q_i = p_{i-1} \cdots p_0 p_{m-1} \cdots p_{i+1} p_i + 1 \bmod N,$$

$$\text{and} \quad \mathrm{WPM2}_{-i}(p_{m-1} \cdots p_1 p_0) = q_{m-1} \cdots q_1 q_0,$$

$$\text{where } q_{i-1} \cdots q_0 q_{m-1} \cdots q_{i+1} q_i = p_{i-1} \cdots p_0 p_{m-1} \cdots p_{i+1} p_i - 1 \bmod N,$$

for $0 \leq i < m$. For example, $\mathrm{WPM2}_{+1}(6) = 1$, for N=3. This network was introduced in [16], where it was shown to be the only network of the five defined here to have the ability to simulate any other interconnection function when the networks are treated as sets of permutations on the integers 0 to N-1. This network is also analyzed in [17].

The *Perfect Shuffle* (*PS*) network consists of the shuffle function defined by:

$$\text{shuffle}(p_{m-1}p_{m-2}\cdots p_1p_0) = p_{m-2}p_{m-3}\cdots p_1p_0p_{m-1}$$

and the exchange function defined by:

$$\text{exchange}(p_{m-1}\cdots p_1p_0) = p_{m-1}\cdots p_1\overline{p}_0.$$

The Omega network is a series of Perfect Shuffles [11]. The PS network is also included in the networks of the Omen and RAP machines [5,8]. The usefulness of this type of network is discussed in [10,15,17,25,26].

The *Illiac* network consists of four functions defined by:

$$\text{Ill}_{+1}(j) = j+1 \bmod N, \qquad \text{Ill}_{-1}(j) = j-1 \bmod N,$$

$$\text{Ill}_{+n}(j) = j+n \bmod N, \qquad \text{Ill}_{-n}(j) = j-n \bmod N,$$

for $0 \leq j < N$, where the integer n is the square root of N. It is the network that was implemented on the Illiac IV. The usefulness of this network is discussed in [4,14,16,17].

The *machine instructions* are those operations that each processor can perform on data in its own memory or registers. The notation $x \leftarrow y$ means the contents of y are copies into x, and $x \leftrightarrow y$ means that x and y swap their contents by using a register for temporary storage. It is assumed that the machine instructions include instructions for moving data among the registers, boolean operations, and comparison (e.g., "less than") operations.

A *masking scheme* is a method for determining which PEs will be active at a given point in time. *Data conditional masks* are the implicit result of executing an "if-the-else" statement in an SIMD machine environment. When an "if-then-else" statement is executed, each PE may be executing it on different data, so the outcome of the conditional may differ from one PE to the next. Each PE will set an internal flag so that it will be active for either the "then" or the "else," but not both. The execution of the "else" block must follow the "then" block; i.e., the two blocks cannot be executed simultaneously (see [21] for details). For example, "*if* A > B *then* C ← A *else* C ← B" will cause each PE to load C with the maximum of A and B; i.e., some PEs will execute "C ← A," and then the rest will execute "C ← B." This type of masking is used in such machines as the Illiac IV [4,12] and PEPE [29]. Other types of masking schemes are discussed in [20].

III. PARTITIONED PARALLEL PROCESSING SYSTEMS

Whether or not all N processing elements of an SIMD machine are required for a given computation is highly task dependent. If only 2^r of the $N(=2^m)$ processing elements are needed, the other $N-2^r$ may be used to perform other tasks. An SIMD machine may be partitioned in two ways: single control partitioning and multiple control partitioning.

Single control partitioning is where one instruction stream is sent to all processors in 2^q disjoint groups of 2^r processors and the other $N-2^{q+r}$ processors are not in use. Thus, a group of N microprocessors could be subdivided into up to 2^{m-r} SIMD submachines of size 2^r where all the submachines follow the same instruction stream, but operate on their own data sets.

A set of N microprocessors could be arranged into a reconfigurable system which could be partitioned into many SIMD machines of different sizes, each with its own control unit (hence, its own instruction stream). This is *multiple control partitioning*, and it requires that the network be partitionable in such a way as to allow each group of processors to operate independently. Single control partitioning is a special case of multiple control partitioning. It is defined separately since it can be used on a "regular" SIMD machine, where there is only one instruction stream, while the multiple form could not. Multiple control partitioning requires the hardware for an independent instruction stream for each SIMD submachine [22,23].

IV. THE ALGORITHMS

The algorithms to perform arbitrary interconnections are based on the idea of sorting destination tags. Each data item to be transferred is associated with a destination tag which is the address of the PE to which the data item is to be sent. Then by sorting the destination tags, moving the associated data items with the tags, arbitrary interconnections can be performed. For simplicity, and without loss of generality, for the rest of the paper it will be assumed that the DTR contents are the destination tags themselves. Thus, the algorithms will just sort the destination tags.

Batcher's bitonic sorting method is described in [1,9,26]. How it can be structured for an SIMD machine is discussed in detail in [19] and is summarized here.

A <u>bitonic</u> <u>sequence</u> is (1) a sequence of numbers $x_0, x_1, \ldots x_{r-1}$, such that $x_0 \geq \ldots x_{k-1} \geq x_k \leq x_{k+1} \leq \ldots x_{r-1}$, for some k, $0 \leq k < r$, or (b) the cyclic rotation of such a sequence. Assume that the data in the DTRs of the N PEs of an SIMD machine form a bitonic sequence when listed in order (from PE_0 to PE_{N-1}), and that $DTR(p_{m-1} \ldots p_1 p_0)$ is the contents of the DTR of PE $p_{m-1} \ldots p_1 p_0$. Then to sort the DTR data do the following:

```
for i = m-1 step -1 until 0 do
    {$DTR(p_{m-1} \ldots p_{i+1} 0 p_{i-1} \ldots p_0) \leftarrow$
        $min(DTR(p_{m-1} \ldots p_{i+1} 0 p_{i-1} \ldots p_0), DTR(p_{m-1} \ldots p_{i+1} 1 p_{i-1} \ldots p_0))$;
     $DTR(p_{m-1} \ldots p_{i+1} 1 p_{i-1} \ldots p_0) \leftarrow$
        $max(DTR(p_{m-1} \ldots p_{i+1} 0 p_{i-1} \ldots p_0), DTR(p_{m-1} \ldots p_{i+1} 1 p_{i-1} \ldots p_0))$}.
```

For each value of i the DTR data of the two PEs whose addresses differ in the i-th bit position are compared. The maximum of each data pair is moved to the PE with a 1 in the i-th position and the minimum to the PE with a 0 in the i-th position. All PEs whose i-th address bit is 0 compute the N/2 minimum in parallel, and then all PEs whose i-th address bit is 1 compute the N/2 maximums in parallel. The interconnection network is used to transfer the DTR data between different PEs.

To sort arbitrary sequences, as opposed to bitonic sequences, m stages of bitonic sorters are needed. The i-th stage acts as 2^{m-i} 2^i-element bitonic sequence sorters. Half of these are ascending sorters, producing ascending sequences, as described above. The other half are descending sorters, producing descending sequences by switching the "min" and "max" functions in the above algorithm. By alternating the ascending and descending bitonic sequence sorters at stage i, $2^{m-(i+1)}$ bitonic sequences of length 2^{i+1} are formed. These bitonic sequences are sorted by stage i+1. An arbitrary sequence sorter is constructed by starting with N/2 alternating ascending and descending 2-element bitonic sorters (simple 2-element comparators), then N/4 alternating ascending and descending 4-element bitonic sorters, etc., ending with a single N-element bitonic sorter (see [9]). Thus, the following instructions would sort the data in the DTRs of the PEs:

```
for j=1 step +1 until m do
 for i=j-1 step -1 until 0 do
  {if j=m or ADDR(j) = 0
    then {$DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0) \leftarrow$
              $\min(DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0), DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0))$;
          $DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0) \leftarrow$
              $\max(DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0), DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0))$}
    else {$DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0) \leftarrow$
              $\max(DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0), DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0))$;
          $DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0) \leftarrow$
              $\min(DTR(p_{m-1}\cdots p_{i+1}0p_{i-1}\cdots p_0), DTR(p_{m-1}\cdots p_{i+1}1p_{i-1}\cdots p_0))$}}.
```

The conditional "if j=m or ADDR(j) = 0" determines which PEs will act as ascending comparators and which will act as desending comparators. To perform the above algorithm on an SIMD machine the interconnection network must be used. For multiple control partitioning, the algorithm must use only the 2^r PEs of a partition to sort the data in that partition to insure the independence between partitions necessary for the multiple control form of partitioning. Recall that single control partitioning is a special case of multiple control partitioning where this independence is unnecessary, since there is a single instruction stream.

Theorem 1: The Cube network can (1) sort an arbitrary sequence of N elements using $m(m+1)/2$ interprocessor data transfers and (2) sort an arbitrary sequences of 2^r elements in a multiple control partitioned environment using $r(r+1)/2$ interprocessor data transfers, $0 \le r < m$.

Proof: The following algorithm directly implements the sorting method described above. The variable "type" specifies whether a comparison should produce an ascending or descending sequence. The A register of each PE is used to store its DTR contents. The DTR contents of the PE whose address differs in the i-th bit position is then received.

```
for j=1 step +1 until m do
  if j=m or ADDR(j)=0 then type ← 0 else type ← 1
  for i=j-1 step -1 until 0 do
    A ← DTR
    $cube_i$
    if type=0
      then if ADDR(i)=0 then DTR ← min(DTR,A) else DTR ← max(DTR,A)
      else if ADDR(i)=0 then DTR ← max(DTR,A) else DTR ← min(DTR,A)
```

The number of interprocessor data transfers used is $m(m+1)/2$. The asymptotic time complexity of the entire algorithm is $O(m^2)$.

Consider any single partition consisting of those PEs whose addresses agree in the m-r high order bit positions. Then if the algorithm above is modified by changing all occurrences of "m" to "r" it will sort the 2^r items from that partition using $r(r+1)/2$ interprocessor data transfers. The asymptotic time complexity of the entire algorithm is $O(r^2)$. To adapt the algorithm to operate when the addresses of the 2^r PEs in a partition agree in any m-r address bit positions, the inner loop controlling i must be modified to step through the r bit positions in which the PEs differ, instead of the low order r positions. This approach works for the multiple control partitioning, because only the PEs in a given partition are involved in the sorting algorithm for that partition.

Theorem 2: The WPM2I network can (1) sort an arbitrary sequence of N elements using $m(m+1)$ interprocessor data transfers, and (2) sort an arbitrary sequence of 2^r elements in a multiple control partitioned environment using $r(r+1)$ interprocessor data transfers, $0 \leq r < m$.
Proof: Follows from Theorem 1, since cube_i is equivalent to the following sequence of instructions (recall the execution of the "then" block precedes the execution of the "else" block of a conditional):

```
$C \leftarrow DTR$
if $ADDR(i)=0$ then $WPM2_{+i}$ else $C \leftrightarrow DTR$; $WPM2_{-i}$; $DTR \leftarrow C$.
```

A formal correctness ~~proof of~~ this equivalence may be found in [18].

Theorem 3: The PM2I network can (1) sort an arbitrary sequence of N elements using m^2 interprocessor data transfers and (2) sort an arbitrary sequence of 2^r elements in a multiple control partitioned environment using $r(r+1)$ interprocessor data transfers, $0 \leq r < m$.
Proof: Part (2) of this theorem follows from Theorem 2, by substituting $PM2_{+i}$ and $PM2_{-i}$ for $WPM2_{+i}$ and $WPM2_{-i}$, respectively. For part (1), cube_i, $0 \leq i \leq m-2$, is equivalent to the sequence of instructions:

```
$PM2_{+i}$;    if $ADDR(i)=0$ then $PM2_{-(i+1)}$ else null;
```

and $\text{cube}_{m-1} = PM2_{-(m-1)}$. A formal correctness proof of these equivalences may be found in [18]. If these equivalences were substituted in the algorithm of Theorem 1 then $m(m+1)-1$ interprocessor data transfers would be used. The number of data transfers can be reduced if PEs (i.e., the ADDR registers) and destination tags are renumbered by the transformation: $p_{m-1}\cdots p_1p_0 \rightarrow p_0p_{m-2}\cdots p_2p_1p_{m-1}$ and the cube_{m-1} and cube_0 functions are switched in the algorithm. Then the $\text{cube}_{m-1} = PM2_{-(m-1)}$ function would be used m times instead of once, and the cube_0 function, which requires two PM2I functions, would be used once instead of m times. Thus, an additional saving of $m-1$ transfers occurs, and the total is m^2.

If the partitioning for part (2) is restricted such that p_{m-1} is not one of the bit positions in which all of the addresses of the PEs in the partition agree, then this technique may be applied. This will reduce the number of interprocessor data transfers to r^2.

Theorem 4: The Perfect Shuffle network can (1) sort an arbitrary sequence of N elements using $(3/2)m^2 - (m/2)$ interprocessor data transfers and (2) sort an arbitrary sequence of 2^r elements in a single control partitioned environment using $m(r-1) + r(r+1)/2$ interprocessor data transfers.
Proof: The algorithm for part (1) is based on Theorem 1, with the following differences: (a) for the first stage the exchange is used instead of cube_0, and for all other stages the cube_i connection is achieved by using $m-i$ shuffles, followed by an exchange, followed by i more shuffles (proved in [18]); and (b) since the data is shuffled $m-i$ times before the exchange, bit position 0 of the PE addresses is examined instead of bit position i to

determine, for a given type, which PE of each pair receives the minimum and which the maximum data element.

```
A ← DTR
exchange
if ADDR(1)=0
   then if ADDR(0)=0 then DTR ← min(A,DTR) else DTR ← max(A,DTR)
   else if ADDR(0)=0 then DTR ← max(A,DTR) else DTR ← min(A,DTR)
for j=2 step +1 until m do
   if j=m or ADDR(j)=0 then type ← 0 else type ← 1
   for i=1 step +1 until m-j do shuffle
   for i=1 step +1 until j do
     { shuffle; A ← DTR; exchange;
       if type=0
          then if ADDR(0)=0 then DTR ← min(A,DTR) else DTR ← max(A,DTR)
          else if ADDR(0)=0 then DTR ← max(A,DTR) else DTR ← min(A,DTR) }
```

This algorithm uses m(m-1) shuffles and m(m+1)/2 exchanges. The asymptotic time complexity of the entire algorithm is $O(m^2)$. An algorithm similar to this is presented by Stone in [26], where the shuffle interconnection was used to interconnect compare-exchange modules.

The algorithm from part (1) can be adapted to satisfy part (2) of this theorem. Assume that the address of all PEs in the same partition agree in the upper m-r bit positions. Then to sort the arbitrary sequences of 2^r elements modify the algorithm by changing "m" to"r" in the control statement for the j loop and in the first statement within the j loop. In the second statement of the j loop, "m-j" remains. This new algorithm uses m(r-1) + r(r+1)/2 data transfers and the asymptotic time complexity of the entire algorithm is O(mr). With the given model of SIMD machines and this approach to sorting, the PS network is constrained to operate in single control mode because the interprocessor data transfers are not contained within each partition.

Theorem 5: The Illiac network can (1) sort an arbitrary sequence of N elements using (25/2)n - 4m -13 interprocessor transfers and (2) sort an arbitrary sequence of 2^r elements in a multiple control partitioned environment using $O(2^{r/2})$ interprocessor data transfers.

Proof: The algorithm for part (1) is based on the algorithm in Theorem 1, using the following substitutions for the the Cube functions.

Case 1: $cube_{m-1}$.

```
   for j=1 step +1 until n/2 do $Ill_{+n}$
```

Case 2: $cube_i$, $m/2 \le i \le m-2$.

```
   B ← DTR
   for j=1 step +1 until $2^i/n$ do $Ill_{+n}$
   B ↔ DTR
   for j=1 step +1 until $2^i/n$ do $Ill_{-n}$
   if ADDR(i)=1 then DTR ← B else null
```

Case 3: $cube_{(m/2)-1}$.

```
   for j=1 step +1 until n/2 do $Ill_{+1}$
   B ← DTR
   $Ill_{-n}$
   if ADDR((m/2)-1)=1 then DTR ← B else null
```

Case 4: $cube_i$, $0 \le i \le (m/2)-2$.

Modify Case 2 by replacing each occurrence of "n" by "1."
A formal correctness proof of these cases may be found in [18]. Since cube_i will be executed m-i times and some cube_i functions require the execution of more Illiac functions than others, a renumbering technique, such as the one described in [27], is employed. If n is an integer, then m is even. If PEs (i.e., ADDR registers) and destination tags are renumbered by the transformation:

$$p_{m-1}\cdots p_2p_1p_0 \rightarrow p_{(m/2)-1}p_{m-1}\cdots p_2p_{(m/2)+2}p_1p_{(m/2)+1}p_0p_{(m/2)}$$

and the Cube functions in the algorithm changed so that for i even cube_i is replaced by $\text{cube}_{(m+i)/2}$ and for i odd cube_i is replaced by $\text{cube}_{(i-1)/2}$, then the number of Illiac functions executed can be greatly reduced. In particular, the Illiac equivalent of $\text{cube}_{(m/2)}$ will be executed m times, cube_0 m-1 times, $\text{cube}_{(m/2)+1}$ m-2 times, cube_1 m-3 times, etc. The number of Illiac functions executed will be: $(25/2)n - 4m - 13$. The asymptotic time complexity of the entire algorithm is $O(n)$. Since the address of the PEs are transformed the conditional statement in Case 2 must use "2i-m" instead of "i," the conditional statement in Case 3 must use "m-1" instead of "(m/2)-1," and the conditional statement in Case 4 must use "2i+1" instead of "i."

This algorithm and renumbering scheme can be adapted for part (2). To simplify the task and allow the use of the techniques just presented assume that (a) all PEs will be renumbered as indicated for part (1) and (b) all PEs in a partition agree in the high order m-r bits of their transformed addresses.

When a partition is performing the sorting algorithm the Illiac equivalents of the Cube functions must be modified as described below to prevent data from being transferred to a PE outside of that partition. (Recall that because of the address transformations p_i is in postion 2i-m for $m/2 \leq i < m$ and in position 2i+1 for $0 \leq i < m/2$.) For cube_i, $m/2 \leq i \leq m-2$ replace "Ill_{+n}" with "if bit 2i-m of [ADDR-(j-1)n] is 0 then Ill_{+n} else null;" and replace "Ill_{-n}" with "if bit 2i-m of [ADDR + (j-1)n] is 1 then Ill_{-n} else null." For cube_i, $0 \leq i \leq (m/2)-2$ replace "Ill_{+1}" with "if bit 2i+1 of [ADDR-(j-1)] is 0 then Ill_{+1} else null," and replace "Ill_{-1}" with "if bit 2i+1 of [ADDR+(j-1)] is 1 then Ill_{-1} else null." Then use the algorithm from Theorem 1, substituting for cube_i, i even, the modified Illiac equivalent of $\text{cube}_{(m+i)/2}$, and for cube_i, i odd, the modified Illiac equivalent of $\text{cube}_{(i-1)/2}$.

The number of interprocessor data transfers used by this new algoritnm is $(14)2^{r/2}-4r-14$ for r even and $r < m-1$, $(19)2^{(r-1)/2}-4r-14$ for r odd and $r < m-1$, and $(19/2)n - 4m - 10$ for $r=m-1$. In all cases the asymptotic time complexity for the entire algorithm is $O(2^{r/2})$. This algorithm may be used for sorting in a multiple control partitioned environment, because only the PEs in a given partition are involved in the sorting for that partition.

V. CONCLUSIONS

Batcher first presented the ideas of bitonic networks and sorting destination tags in [1]. Stone then used this notion with N/2 compare-exchange modules linked by a shuffle function [26]. Orcutt presented an algoritnm for the Illiac IV network to implement bitonic sorting on an SIMD machine using 2nm-6m+16n-16 interprocessor data transfers [13,14]. In [19], the way in which bitonic sorters could be implemented on SIMD machines was analyzed in detail and some preliminary results relating to the Cube, PM2I, WPM2I, and PS networks were discussed. In [27], Thompson and Kung used a modified Illiac IV network to implement bitonic sorting using 14n-4m-14 interprocessor data transfers.

In this paper, algorithms to demonstrate the universality of the Cube, WPM2I, PM2I, PS, and Illiac networks were given. In addition, two methods of partitioning a reconfigurable multimicroprocessor parallel processing system into SIMD machines were defined, and the universality of each of the five networks in partitioned systems was evaluated. The algorithms presented can be implemented on any computer system satisfying the few assumptions made.

Based just on sorting speeds, the Cube network is best. If it is desirable to use fewer than m interconnection functions, an augmented PS network can be used with less than a factor of two degradation for sorting N items and a factor of 2m/r for sorting 2^r items. If the PS network is augmented by a shuffle-exchange function, i.e., a function which does the equivalent of a shuffle followed by an exchange, then the number of data transfers to sort can be reduced to m(r-1)+1, $1 \leq r \leq m$. However, due to the structure of the shuffle interconnection, neither this augmented PS network nor the original PS network can operate in a multiple control partitioned system using the SIMD machine model presented. The other networks, requiring more interconnection functions, were shown to be universal even in a multiple control environment.

Acknowledgements. The author thanks L. J. Siegel, J. D. Ullman, and H. S. Stone for their comments and suggestions.

REFERENCES

1 Batcher, K. E., "Sorting networks and their applications," 1968 Spring Joint Computer Conf. *AFIPS Proc.* 32 (1968), 307-314.

2 Batcher, K. E., "The multi-dimensional access memory in STARAN," *IEEE Trans. Comp.* C-26 (Feb. 1977), 174-177.

3 Batcher, K. E., "The flip network in STARAN," *1976 Int'l. Conf. Parallel Processing* (Aug. 1976), 65-71.

4 Bouknight, W. J., et. al., "The Illiac IV system," *Proc. IEEE* 60, 4 (Apr. 1972), 369-388.

5 Couranz, C. R., Gerhardt, M. S., Young, C. J., "Programmable RADAR signal processing using the RAP," *1974 Sagamore Comp. Conf. Parallel Processing* (Aug. 1974), 37-52.

6 Feng, T., "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comp.* C-23 (Mar. 1974), 309-318.

7 Flynn, M. J., "Very high-speed computing systems," *Proc. IEEE,* Vol. 54 (Dec. 1966), pp. 1901-1909.

8 Higbie, L. C., "The Omen computer: associative array processor," *Compcon 72,* IEEE Computer Soc. Conf. (Sep. 1972), 287-290.

9 Knuth, D. E., *The Art of Computer Programming: Vol. 3 Sorting and Searching,* Addison-Wesley (1973), section 5.3.4.

10 Lang T., Interconnections between processors and memory modules using the shuffle-exchange network," *IEEE Trans. Comp.* C-25 (May 1976), 496-503.

11 Lawrie, D. H., "Access and alignment of data in an array processor," *IEEE Trans. Comp.* C-24 (Dec. 1975), 1145-1155.

12 Lawrie, D. H., et.al., "Glypnir - a programming language for Illiac IV, *Comm. ACM* 18, 3 (Mar. 1975), 157-164.

13 Orcutt, S. E., Computer organization and algorithms for very-high speed computation, Stanford University CS Dept. Ph.D. Thesis (Sep. 1974).
14 Orcutt, S. E., "Implementation of permutation functions on Illiac IV - type computers," IEEE Trans. Comp. C-25 (Sep. 1976), 929-936.
15 Pease, M. C., "The indirect binary n-cube microprocessor array," IEEE Trans. Comp. C-26 (May 1977), 458-473.
16 Siegel, H. J., "Analysis techniques for SIMD machine interconnection networks and the effects of processor address masks," IEEE Trans. Comp. C-26 (Feb. 1977), 153-161.
17 Siegel, H. J., "Single instruction stream-multiple data stream machine interconnection network design," 1976 Int'l. Conf. Parallel Processing (Aug. 1976), 273-282.
18 Siegel, H. J., SIMD machine interconnection network design, Princeton University EE/CS Department, TR-198 (Jan. 1976).
19 Siegel, H. J., "The universality of various types of SIMD machine interconnection networks," 4th Symp. Comp. Arch. (Mar. 1977), 70-79.
20 Siegel, H. J., "Controlling the active/inactive status of SIMD machine processors," 1977 Int'l. Conf. Parallel Processing (Aug. 1977), pg. 183.
21 Siegel, H. J., Methods for determining the active/inactive status of single instruction stream - multiple data stream machine processors, Purdue University Electrical Engineering School, TR-EE 77-25 (May 1977).
22 Siegel, H. J., "Preliminary design of a versatile parallel image processing system," Third Biennial Conf. Computing in Indiana (Apr. 1978), 11-25.
23 Siegel, H. J., Mueller, P., Smalley, H., "Control of a partitionable multimicroprocessor system," 1978 Int'l. Conf. Parallel Processing (Aug. 1978).
24 Siegel, H. J., Smith, S. D., "Study of multistage SIMD interconnection networks," 5th Symp. Comp. Arch. (Apr. 1978), pp. 223-229.
25 Smith, S. D., Siegel, H. J., "Recirculating, pipelined, and multistage SIMD interconnection networks," 1978 Int'l. Conf. Parallel Processing (Aug. 1978).
26 Stone, H. S., "Parallel processing with the perfect shuffle," IEEE Trans. Comp. C-20 (Feb. 1971), 153-161.
27 Thompson, C. D., Kung, H. T., "Sorting on a mesh-connected parallel computer," Comm. ACM 20,4 (Apr. 1977), 263-271.
28 Wester, A. H., "Special features in SIMDA," 1972 Sagamore Comp. Conf. (Aug. 1972), 29-40.
29 Wilson, D. C., "The PEPE support software system," Compcon 72, IEEE Comp. Soc. Conf. (Sep. 1972), 61-64.

A COMPUTER ARCHITECTURE FOR PARALLEL PROCESSING*

T. N. Mudge
Dept. ECE, Univ. Michigan, Ann Arbor, MI 48109.

SUMMARY

This paper briefly presents some ideas arising out of a study to determine the requirements of a computer architecture that exploits potential parallelism in its source programs.

The architecture is organized to execute a base language which obeys the single assignment (SA) rule (each variable in a program may be assigned at most once). It has been shown in [U] that programs written in an SA language exhibit maximal parallelism during their execution, provided they have sufficient resources. Unfortunately, several obstacles need to be overcome to make SA architectures practical. These are: Implementing iterative procedures (e.g. DO loops) in the face of the SA rule, coping with possible unbounded memory requirements, and handling computed references. A solution to the first of these problems exist if iteration in the base language is represented in a recursive fashion (see [M1] and [U] for details). A solution to the second problem is given in [M1] by associating a read count with each variable (see also [M2] for a proof of the validity of this procedure). Finally, the last of these problems is resolved in [M2] using a technique similar to that proposed in [A] for data flow machines.

To take advantage of the maximal parallelism present in programs written in the base language, the architecture is organized as a high bandwidth memory with multiple instruction-set processors. A result-forwarding scheme between the instruction-set processors is employed to reduce the bandwidth requirements of the memory. Source programs are translated into the SA base language by a translator (written in the base language), then loaded and then run. All the system programs are expressed in the base language, so they may take advantage of an architecture organized to execute the SA base language efficiently. Note that the translator can be regarded as an optimizer, which is aimed at a particular type of machine organization. Further optimization is possible, if potential functional parallelism is exploited.

[A] Ackerman, W. B.,"A Structure Processing Facility for Data Flow Computers", Proc. 1978 Internat. Conf. On Parallel Processing, Aug. 1978.

[M2] Mudge, T. N., "A Data Driven Computer Architecture", Proc. 1978 Conf. on Information Sciences and Systems, pp. 365-370, March 1978.

[M2] Mudge, T. N., and W. T. Liu,"Memory Management Procedures for a Single Assignment Programming Language",Systems Eng. Lab. Report, Dept. ECE, Univ. Michigan, (to appear).

[U] Urschler, G., "The Transformation of Flow Diagrams into Maximally Parallel Form", Sagamore Comp. Conf. on Parallel Processing, pp. 38-46, March 1973.

---

*This work was supported by the National Science Foundation under grant ENG-78-05779

NEW UPPER BOUNDS ON THE NON-ASYMPTOTIC COMPLEXITY OF MATRIX MULTIPLICATION

ROBERT L. PROBERT,
Department of Computational Science,
University of Saskatchewan,
Saskatoon, Sask., Canada.

ABSTRACT

A brief overview is given of algorithms and known upper bounds on the complexity of multiplying two n x n matrices by {0,1,-1} - bilinear chains. A problem decomposition technique is introduced and then applied to yield a methodology for generating new {0,1,-1} - bilinear chains which are the "fastest" known for "small" values of n ($n \le 40$ approximately). Extensions and limitations of this non-asymptotic approach are presented. Finally, tables of algorithms and costs are given which incorporate the new algorithms just discovered by V. Pan which lead to an improvement of Strassen's rate of growth.

INTRODUCTION

Since Strassen [16] demonstrated that two n x n matrices could be multiplied in approximately $n^{2.81}$ multiplications rather than the n^3 multiplication steps of the classical algorithm by discovering a surprisingly simple algorithm called a bilinear chain to multiply 2 x 2 matrices using only seven nonscalar multiplications, many researchers have studied the arithmetic complexity of matrix multiplication [1-18]. In particular, several of these have attempted to derive a multiplicative cost function M(n) for n x n matrix multiplication by such non-commutativity-exploiting, simple bilinear chains (since these algorithms are the least powerful of any class of straight-line polynomial programs, practical algorithms which might exploit field dependencies, or commutativity could only perform better) such that for even one value of n, $\log_n M(n) < \log_2 7 \simeq 2.81$. The goal was to apply such an algorithm recursively to obtain a better asymptotic rate of growth than Strassen's scheme. For many years, in fact until just before the time of writing this paper, no such improvement was found (see the final section for a discussion of the new result). And yet, the best lower bound for this problem to date was $2n^2 - 1$ [2]. Since it had been shown in several papers that matrix multiplication was of the same order of multiplicative complexity as matrix inverse, solving linear systems, and computing the transitive closure of a directed graph, considerable importance was attached to bringing the lower and upper bounds closer together or at least to gaining more insight into the essential complexity of matrix multiplication.

Several new algorithms were discovered in attempts to derive useful representations of the problem [4,10], but since these matched the behaviour of Strassen's algorithm for n=2 and were comparatively inefficient for n>2, no significant progress towards finding new upper bounds was achieved other than a slight decrease in the bound for 3x3 multiplication from 24 to 23. However, in order to better Strassen-like algorithms by a method based on n=3, a 21-multiplication method is required.

In this paper, we do not attempt to better Strassen's algorithm's rate of growth, but rather to unify, simplify and improve on the state of the knowledge about very simple Strassen-like schemes for small ($n \le 40$) matrix orders. This approach in a preliminary form enabled us to show [6,7] that an intelligent implementation of Strassen's scheme leads to a crossover point of n=14 for the arithmetic cost of fast algorithms versus the classical algorithm. The fact that the crossover point is so low may be surprising to several who may have peremptorily ruled out Strassen-like algorithms as being impractical.

To accomplish this, in the next two sections we acquaint the reader with the terminology and results of the area and then introduce a problem decomposition or operator-oriented approach. The philosophy of this methodology is simply to use combinations of the various known algorithms as efficiently as possible, i.e., given a library of best algorithms known, use algorithm transformation operators to generate new algorithms from old. In the final section, we present a brief discussion of the effectiveness and limitations of this approach, using the asymptotically better algorithm just announced by V. Pan [11] as an example. Finally, tables are given which summarize the "fastest" (of least multiplicative cost) algorithms for non-asymptotic matrix orders incorporating this recent development.

BASIC DEFINITIONS AND RESULTS

A matrix multiplication algorithm for computing the product $X_{mxn} * Y_{nxp} = C_{mxp}$ (which we refer to as an (m,n,p) product) is a sequence of steps of the form $f_i \leftarrow f_j$ "op" f_k where "op" ε {'+', '-', '*'} and f_j, f_k are elements of X, Y, a scalar ring K, or the results of previous steps.

A matrix multiplication algorithm for computing X * Y = c is called a K - bilinear chain if, in every step of the form $f_i \leftarrow f_j \times f_k$ where neither f_j nor f_k is in K, f_j is a K-linear combination of elements from A and f_k is a K-linear combination of elements from B.

In this paper, all algorithms considered will be {0,1,-1} - bilinear chains, i.e., K = {0,1,-1}.

The multiplicative cost of such an algorithm is the number of steps of the form $f_i \leftarrow f_j * f_k$ where neither of f_j or f_k is in K.

We will make use of the following result [12, 2, 5]:

Theorem (Computational Duality):

Given an algorithm α for multiplying (m,n,p) products where α does not exploit commutativity of multiplication of matrix elements (such algorithms properly include K-bilinear chains), and any permutation (u,v,w) of (m,n,p), one can construct an algorithm α' directly from α which uses exactly the same number of multiplication steps as α.

This duality theorem will be applied to the {0,1,-1} bilinear chain for (m,2,n) products due to Hopcroft and Kerr [8] which uses $\left[\frac{\max(m,n) + 3mn}{2}\right]$ multiplication steps. This yields any desired member of the class of {0,1,-1} - bilinear chains which are duals of a chain for (m,2,n) products. We will need this result to define one of the operators in the next section. For example, since Hopcroft and Kerr have an algorithm which computes (5,2,5) products in 40 multiplication steps, by computational duality, we can construct a dual algorithm which computes (2,5,5) products in the same number of multiplication steps.

The rationale for attempting to discover algorithms which do not exploit commutativity is that such algorithms can operate on submatrices as elements. More formally, the following was observed by Winograd and Strassen [16, 18]:

Lemma (Recursive Implementations):

If there exists an algorithm which computes (ℓ,ℓ,ℓ) products in c multiplication steps without exploiting commutativity of multiplication then there exists an algorithm which again does not exploit commutativity to compute (k,k,k) products where k is some power of ℓ in $k^{\log_\ell c}$ multiplication steps.

Strassen [16] showed that the cost for any n was bounded above by $4.7n^{\log_2 7}$ and this constant was lowered by P. Fischer, and the author in [6,7] to $3.912\ n^{\log_2 7}$. As noted in the introduction, however, this asymptotic viewpoint can be misleading for "small" values of n.

THE PROBLEM DECOMPOSITION APPROACH

In this section we present five algorithm operators (originally defined in [15]) which can be combined in various ways to yield composite algorithms for matrix multiplication. Each operator transforms an algorithm γ_1 for $n_1 \times n_1$ matrix multiplication, denoted $\gamma_1(n_1)$, to an algorithm $\gamma_2(n_2)$. Thus, any composite algorithm of the form $\gamma_1\gamma_2\ldots\gamma_k$ describes a feasible decomposition of a given matrix multiplication problem into multiplication subproblems for matrices of orders $n_1, n_2, \ldots, n_k$.

Assume we are given an algorithm $\gamma(n_1)$ which multiplies arbitrary $n_1 \times n_1$ matrices using c_1 non-scalar multiplications. Then for each operator F, Fγ yields an algorithm for multiplying $n_2 \times n_2$ matrices at cost c_2. The relationships between n_1, n_2 and c_1, c_2 are as follows:

Operator	n_2	c_2
W	$2n_1$	$7c_1$
L	$3n_1$	$23c_1$
B	$n_1 + 1$	$c_1 + 3n_1^2 + 3n_1 + 1$
R	$n_1 + 2$	$c_1 + \frac{9n_1^2 + 21n_1 + 16}{2}$
Z	$n_1 - 1$	c_1

Description of Operators:

W is a Strassen-like [16] operator for multiplying 2n x 2n matrices as 7 products of n x n matrices according to the identities presented in [13] and due to Winograd.

L is a generalization due to Lewis [10] of an algorithm discovered by Laderman [9] for multiplying 3 x 3 matrices using only 23 nonscalar multiplications.

Z is unsophisticated zero-padding. An algorithm γ for multiplying matrices of order n + 1 is used for an order n matrix multiplication problem. Non-scalar multiplication steps of γ are counted even if one of the operands is zeroed as a result of the zero-padding.

B is the well-known block multiplication scheme for multiplying matrices of order n + 1 by using γ to compute the product of the upper left quadrant n x n submatrices of the given matrices and the classical algorithms to compute all residual degenerate products.

R is a new operator constructed from γ and dual algorithms [12] of Hopcroft and Kerr's efficient algorithm for multiplying an (m, 2, n) matrix product [15]. If XY = C is the product matrix of order n + 2, R partitions C as C_1 (2 x (n+2)), C_2 (n x 2), and C_3 (n x n). Then, C_1 is computed as a (2, n+2, n+2) product (a dual of (n+2, 2, n+2)), C_2 as (n, n+2, 2) product (a dual of (n, 2, n+2) and C_3 as the sum of an (n, n, n) product (computed using γ) and an (n, 2, n) product.

To illustrate how these operators can be used to generate efficient {0, 1, -1} - bilinear chains, consider the problem of computing the product of two 5 x 5 matrices.

The classical algorithm uses 125 multiplications.

Adding a row and column of zeros to each matrix, then applying a Strassen-like algorithm to reduce the problem to seven multiplications of 3 x 3 matrices, and finally using the Lewis algorithm to compute the (3, 3, 3) products in 23 multiplications uses a total of 7 x 23 or 161 multiplications (without trying to ignore zeros introduced by the original zero-padding).

Thus, the problem decomposition ZWL (read from the left) is not efficient with respect to multiplicative complexity.

The decomposition consisting of the block multiplication operator followed by a recursive application of the Strassen-like operator, denoted BWW or BW^2 uses 110 multiplications.

But the "fastest" algorithm known for multiplying 5 x 5 matrices due to this author [15] corresponds to the decomposition RL and uses 103 multiplications. Of course, this does not represent an asymptotic improvement over Strassen's algorithm since none of the other operators have associated cost increments which improve on operator W ($\log_5 103$ is approximately 2.88 versus 2.81). However, this example indicates the approach used in the rest of this section to generate new upper bounds on matrix multiplication for small (non-asymptotic) matrix orders.

These upper bounds, corresponding algorithms, and relativized logarithms are given for a few representative values in Table 1. Successive repetition of an operator is indicated by an exponent.

It should be noted that these costs are smaller and in most cases, considerably smaller than those of algorithms due to Fiduccia [4], Lewis [10], and Sykora [17], even with the improvements made by Sykora to his algorithm for 5 x 5 multiplication to obtain a cost of 105 multiplications. Of course, these authors were more interested in presenting new characterizations of the matrix multiplication problem than in producing algorithms which minimized the number of multiplications used.

It is possible to obtain some intuition about the optimal problem decomposition for a given set of operators by observing the relativized logarithmic behaviour of these algorithms.

For example, given a problem of computing (n, n, n) products we are interested in minimizing the cost c(n) or equivalently in minimizing the value of $\log_n c(n)$ (since n is fixed). Employing the Z operator is comparable to computing $\log_n c(n)$ as

$$\log_n c(n+1) = \log_{n+1} c(n+1) \quad /\log_{n+1}(n)$$

whereas employing the β operator first is comparable to computing it as

$$\log_n(c(n-1) + 3(n-1)^2 + 3(n-1) + 1)$$
$$= \log_{n-1}(c(n-1) + 3n^2 - 3n+1) \quad /\log_{n-1}(n)$$

If approximations to $\log_{n+1} c(n+1)$ and $\log_{n-1} c(n-1)$ are known, the smaller approximation should be favoured in general. For example, for n = 13, $\log_{12} c(12) \cong 2.828$ and $\log_{14} c(14) \cong 2.863$. Thus, we would expect $\beta\ \gamma(12)$ to beat $Z\ \gamma(14)$ which it does.

Similar expansions can be analyzed for comparisons among the other operators, as well as for the implicit operators given by any factorization of a given matrix order. Suppose $n = ab$. Then, employing an imbedding consisting of computing (a, a, a) products where each element is a b x b submatrix corresponds to computing $\log_n c(n)$ as

$$\log_{ab}(c(a)c(b)) = \frac{\log_a c(a)}{1 + \log_a b} + \frac{\log_b c(b)}{1 + \log_b a}$$

which can be informative if approximations to $\log_a c(a)$ and $\log_b c(b)$ are known. For example, if $n = 15$, since $\log_3 23 \cong 2.85$ and $\log_5 103 = 2.88$, we would expect $\log_{15}(23 \mathrm{x} 103)$ to approximate 2.87, which it does. In fact, this last observation corresponds to an analytical proof that the only such imbedding operators which need be considered for a non-asymptotic analysis like the one in this paper are W and T.

EXTENSIONS AND LIMITATIONS

One of the strengths of this class of algorithms ($\{0,1,-1\}$-bilinear chains) is that it is the least powerful class of reasonable algorithms for computing sets of bilinear forms and thus any upper bounds obtained can only be improved upon by more "realistic" algorithms. And yet, asymptotically speaking [14], more powerful classes of algorithms cannot reduce the multiplicative or even total arithmetic complexity.

However, in a non-asymptotic analysis it is interesting to note that algorithms which exploit commutativity such as Winograd's [18] fast inner product algorithm with its $\frac{n^3}{2} + n^2$ multiplication steps beats even W^t for $n = 2^t$ and $t < 5$. Therefore, such algorithms are interesting from a non-asymptotic (even from a practical) viewpoint.

Another extension of $\{0,1,-1\}$ - bilinear chains is obtained by enlarging the set K to include $\frac{1}{2}$. De Pillis [3] gives an algorithm to compute (6, 2, 6) products faster than Hopcroft and Kerr and Brockett and Dobkin [1] give a general technique to compute (n, 2, n) products faster than Hopcroft and Kerr for $n > 196$. By the way, the Brockett-Dobkin algorithm computes (n, p, n) products in approximately n^2 multiplication steps if $p < \log_2 n$. For a non-asymptotic analysis, algorithms such as that of de Pillis and their duals could be incorporated into the operator R to make it more efficient provided we allow any accompanying extensions of the underlying algebra.

A refinement of the Z operator to a more "knowledgeable" operator K is also possible. Algorithms corresponding to problem decompositions involving knowledgeable zero-padding would note when one of the operands in a multiplication is 0 and avoid performing the multiplication operation. This would save $4n+1$ multiplication steps for $n = 2^t - 1$. In turn, this would affect the upper bounds listed in Table 1 and can even alter some decomposition strategies. For example, the decomposition for n = 15 would have to be changed from TRT to KW^4. The savings can be considerably greater for other values of n. Also, if zero operations are avoided, a new operator C corresponding to the classical algorithm would need to be included in the library of operators. This follows from the observation that for quite sparse matrices, the classical algorithm can use fewer non-zero multiplications than Strassen-like algorithms. For example, if the algorithm is allowed to choose between W and C at the base level of multiplications, only 2308 multiplication steps need be employed to multiply matrices of order 15.

A major limitation of this operator approach may be that it does not take account of global symmetries very well if at all. In particular, the author feels that optimal algorithms for matrix multiplication must be tailor-made for each matrix order. We will now present some evidence for this opinion.

If we were attempting to better the Strassen rate of growth (with non-commutative algorithms, of course), we might attempt to match the relativized logarithmic behaviour of W at its best, namely for n = 2, and then try to improve on this value as n increases. This requirement can be shown to eliminate "natural" cost functions of the form $n^3 - \frac{n^2}{4}$, $\frac{n^3}{2} + \frac{3n^2}{2}$, and $\frac{n^3}{3} + 6n^2 - \frac{4n}{3}$. This is obviously not the approach employed by Pan [11] since the last function is the cost function of an algorithm he has just announced which beats the relativized logarithmic behaviour of Strassen's algorithm at several values of n, minimized at n = 70. Pan also found another algorithm which performed better non-asymptotically than this algorithm, but which does not beat Strassen. This is support for our view that non-composite algorithms favour only particular matrix orders and that the problem decomposition approach has to be used to "spread" the benefits over matrix multiplication problems of all orders.

We have extended the non-asymptotic analysis of Pan [11] for operators corresponding to his algorithms. We denote the asymptotically oriented operator A and the non-asymptotically oriented operator N. Matrix orders, problem decompositons, and new upper bounds are given in Table 2 below.

ACKNOWLEDGEMENTS

The author grafefully acknowledges the insights obtained or solidified in discussions with Patrick Fischer; the original notion of problem decomposition was developed jointly with him and his viewpoint was largely responsible for the operator-oriented approach of this and other papers. This work was partially supported by NRC grant A8982.

REFERENCES

[1] R.W. BROCKETT and D.P. DOBKIN, On the number of multiplications required for matrix multiplication, SIAM J. on Comput. 5, 4 (1976) pp. 624-628.

[2] D.P. DOBKIN, On the arithmetic complexity of a class of arithmetic computations, Research Report 23, Yale University 1973.

[3] J. DE PILLIS, Schemes for fast matrix multiplication, personal communication 1976.

[4] C.M. FIDUCCIA, On obtaining upper bounds on the complexity of matrix multiplication, Complexity of Computer Computations, ed. R.E. Miller and J.W. Thatcher, Plenum Press, New York (1972) pp. 31-40.

[5] C.M. FIDUCCIA, On the algebraic complexity of matrix multiplication, Thesis, Brown University, 1973.

[6] P.C. FISCHER, Further schemes for combining matrix algorithms, Proc. of Second Symp. on Automata, Languages and Programming, Lecture Notes in Computer Science 14, ed. G. Goos and J. Hartmanis, Springer-Verlag (1974), pp. 428-436.

[7] P.C. FISCHER and R.L. PROBERT, Efficient procedures for using matrix algorithms, ibid. (1974) pp. 413-427.

[8] J.E. HOPCROFT and L.R. KERR, On minimizing the number of multiplications necessary for matrix multiplication, Soc. of Ind. and Appl. Math. (SIAM) J. Appl. Math. 20 (1971) pp. 30-36.

[9] J.D. LADERMAN, A non-commutative algorithm for multiplying 3 x 3 matrices using 23 multiplications, Bull. of the A.M.S., Vol. 82, Num. 1 (1976).

[10] J. LEWIS, An algorithm to compute the product of two n x n matrices using $n^3-(n-1)^2$ multiplications, personal communication, 1976.

[11] V. PAN, An introduction to the trilinear technique of aggregating, uniting and cancelling and applications to constructing fast algorithms for matrix operations, personal communication (1978) (to appear in Proc. of 19th FOCS Symp).

[12] R.L. PROBERT, On the complexity of symmetric computations, Canadian J. of Op. Res. and Inf. Processing (INFOR), Vol. 12, No. 1 (1974) pp.71-86.

[13] R.L. PROBERT, On the additive complexity of matrix multiplication, SIAM Journal on Computing, Vol. 5, No. 2 (1976) pp. 187-203.

[14] R.L. PROBERT, Commutativity, non-commutativity and bilinearity, Info. Proc. Lett., 5, 2 (1976) pp. 46-49.

[15] R.L. PROBERT, On the composition of matrix multiplication algorithms, Proc. of Sixth Manitoba Conf. on Num. Math. and Computing, Congressus Numerantium 18 (1977) pp. 357-366.

[16] V. STRASSEN, Gaussian elimination is not optimal, Numer. Math., 13

(1969) pp. 354-356.

[17] O. SYKORA, A fast non-commutative algorithm for matrix multiplication, Proc. of the Sixth Symp. on Math. Foundations of Computer Science Lecture Notes in Comp. Science 53 ed. G. Goos and J. Hartmanis, Springer-Verlag (1977) pp. 504-512.

[18] S. WINOGRAD, On the number of multiplications necessary to compute certain functions, Comm. on Pure and App. Math., 23 (1970) pp. 165-179.

n	cost	decomposition	relativized logarithm
2	7	W	2.81
3	23	L	2.85
4	49	WW	2.81
5	103	RL	2.88
6	161	WL	2.84
7	276	RRL	2.89
8	343	WWW	2.81
9	529	LL	2.85
10	721	WRL	2.86
11	996	RLL	2.88
12	1127	WWL	2.83
13	1596	BWWL	2.88
14	1909	RWWL	2.86
22	6972	WRLL	2.86
23	7889	ZWWWL	2.86
24	7889	WWWL	2.82
25	9690	BWWWL	2.85
31	16807	ZWWWWW	2.83
33	19976	BWWWWW	2.83
34	21759	RWWWWW	2.83
35	25330	BRWWWWW	2.85
36	25921	WWLL	2.84
40	35329	WWWRL	2.84

Table 1. New Upper Bounds, Problem Decompositions, and Relativized Logarithms for nxn Matrix Multiplication for Small n.

n	cost	decomposition	relativized logarithm
10	710	N	2.85
14	1792	N	2.84
18	3618	N	2.83
19	4545	BN	2.87
20	4870	N	2.83
21	6138	BN	2.86
22	6380	N	2.83
26	9380	A	2.82
27	11984	ZA	2.85
28	11984	A	2.82
29	14360	ZA	2.84
30	14360	A	2.81
34	19990	A	2.81
35	23280	ZA	2.83
36	23280	A	2.81

Table 2. New Upper Bounds Incorporating Pan's New Algorithms A and N (values are omitted for which Table 1 cost is unaffected).

Note: WA(100) costs 2752400 (exponent 2.80)
A(200) costs 2906400 (exponent 2.81)

COMPUTING EXTREMAL AND APPROXIMATE DISTANCES IN A GRAPH HAVING UNIT COST EDGES

KELLOGG S. BOOTH
Department of Computer Science
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

RICHARD J. LIPTON
Department of Computer Science
Yale University, New Haven, Connecticut 06520

ABSTRACT

Using binary search and a Strassen-like matrix multiplication algorithm we obtain efficient algorithms for computing the diameter, radius, and other distance-related quantities associated with undirected and directed graphs having unit cost (unweighted) edges. Similar methods are used to find approximate values for the distances between all pairs of vertices, and the exact distances if the graph satisfies certain regularity conditions.

INTRODUCTION

We consider the problem of computing distances in directed and undirected graphs whose edges have unit costs. We will use standard notation from graph theory [1,4]. We assume that a graph $G = (V,E)$ is connected (strongly connected if it is directed), that its vertex set is $V = \{1, 2, \cdots, n\}$, and that its edge set E has e elements. We develop algorithms for computing the diameter, radius, and other distance-related quantities associated with the graph. Our algorithms employ binary search techniques and use fast matrix multiplication routines to achieve asymptotic efficiency. Using similar methods we develop algorithms for computing approximate distances between all pairs of vertices. Finally, we discuss a class of graphs for which the exact distances can be easily computed.

The *distance* between two vertices i and j is the number of edges in a shortest path connecting them. This distance is denoted by $d(i,j)$. The *eccentricity* of a vertex is the greatest among all of its distances to other vertices in the graph. We denote the eccentricity of a vertex by

$$e(i) = \max\{d(i,j) \mid j \in V\}$$

The *diameter* and *radius* of a graph G are the largest and smallest eccentricities of its vertices:

$$diameter(G) = \max\{e(i) \mid i \in V\}$$

$$radius(G) = \min\{e(i) \mid i \in V\}.$$

Vertices of maximum eccentricity are on the *periphery* of G while those of minimum eccentricity are in the *center* of G.

$$periphery(G) = \{i \mid e(i) = diameter(G)\}$$

$$center(G) = \{i \mid e(i) = radius(G)\}$$

A *pair of peripheral points* consists of two vertices whose distance apart is equal to the diameter of the graph. A shortest path between a pair of peripheral points is called a *geodesic*.

For our purposes a graph will be represented by its *adjacency matrix*, $A = (a_{i,j})$, which is the $n \times n$ matrix defined by

$$a_{i,j} = \begin{cases} 1, \text{ if there is an edge } \{i,j\} \in E \\ 0, \text{ if there is no edge } \{i,j\} \in E. \end{cases}$$

The *distance matrix* for G is the $n \times n$ matrix $D = (d_{i,j})$ in which $d_{i,j}$ is the distance between i and j. For directed graphs we define similar notions and consider the obvious in- and out-generalizations for eccentricity, diameter, radius, periphery, and center. Unless otherwise noted we will state our results for undirected graphs and imply the corresponding results for directed graphs.

Floyd's shortest path algorithm computes the distance matrix of a graph in $O(n^3)$ steps [2]. Fredman [3] has an algorithm whose running time is a bit less, $O(n^3(\log\log n/\log n)^{1/3})$, but Floyd's version is the practical choice for real applications.

Both of these algorithms compute the distance matrix of a graph having arbitrary non-negative costs on the edges. Our graphs have unit cost edges. Although this assumption would appear to make the problem easier, we know of no better way of finding the distance matrix for unit cost edges than to apply a general algorithm designed to handle arbitrary cost edges.

In some applications we do not need to know the entire distance matrix. Knowledge of extremal distances such as the diameter or radius may suffice. Or we might need only a list of the peripheral or central vertices. If the graph has arbitrary costs on its edges we see no shortcuts, but for unit cost edges we show how to compute these quantities in $O(M(n)\log n)$ steps, where $M(n)$ is the number of steps required to multiply two $n \times n$ matrices. If we are interested in the entire distance matrix but not in the exact values we show how to obtain, with relative error at most $\epsilon > 0$, the approximate distances between all pairs of vertices in $O(M(n)(\log n)^2)$ steps.

Throughout our discussion we will let $M(n)$ stand for the complexity of $n \times n$ matrix multiplication and we will express all of our upper bounds in terms of this quantity. Thus the results given here will automatically improve whenever a new upper bound is shown for matrix multiplication. Strassen's algorithm provides an $O(n^{\log_2 7})$ upper bound [8]. Pan [6] has recently announced a slightly better asymptotic bound of $O(n^{\log_{64} 111,872})$. Numerically this is $O(n^{2.795})$.

COUNTING WALKS

The problem of computing a graph's diameter can be viewed as finding the smallest integer k such that every pair of vertices is connected by a path whose length does not exceed k. Similarly a graph's radius is the smallest integer k such that some vertex is connected to all other vertices by paths whose lengths do not exceed k. If we can efficiently test these two conditions for any k we can find the minimum k quickly using a binary search. Since both the diameter and the radius are bounded above by n, at most $\log n$ values of k need be tested in order to find the diameter or radius.

Directly deciding the existence of paths of length k between all pairs of vertices might be as hard as computing the entire distance matrix. It seems easier to test a slightly different but closely related condition.

A *walk* between two vertices i and j is a sequence of vertices beginning at i and ending at j having the property that successive pairs of vertices are adjacent in G. In general a walk is not a path because vertices, even edges, may be repeated. But a shortest walk is always a shortest path, and *vice versa*, since otherwise the walk or path could be made even shorter by eliminating the edges connecting two instances of a repeated vertex.

For our purposes walks have one major advantage over paths: the existence of a path of length k from i to j usually tells nothing about the existence of longer paths between i and j, but a walk of length k can always be extended to a walk of length $k+2$ simply by repeating any edge twice, once in each direction. Notice that there might not be a walk of length $k+1$, however, because it is not always possible to add a single edge to a walk without changing its endpoints.

Using this fact and our observation that shortest paths are the same as shortest walks we can state the following characterization for the diameter of a graph.

LEMMA 1: A graph G has diameter k iff
(1) between every pair of vertices there exists a walk of length $k-1$ or k, and
(2) between at least one pair of vertices walks of length $k-2$ and $k-1$ do not exist.

We check for the existence of walks of both even and odd length to avoid the possibility of having the lengths of all walks between a particular pair of vertices be of the same parity, in which case there could fail to be a walk whose length exactly equalled the diameter. The extra check for a walk of length one less than the diameter alleviates this problem because a walk whose length is exactly the diameter or exactly one less than the diameter is guaranteed.

We state a similar characterization for the radius of a graph.

LEMMA 2: A graph G has radius k iff
(1) for some vertex there exist walks of length k or $k-1$ to all vertices, and
(2) for all vertices there is at least one other vertex to which walks of length $k-2$ and $k-1$ do not exist.

To implement an efficient test for these two conditions we cite a well-known result relating walks of length k in a graph to the kth power of the adjacency matrix of the graph [1,4].

LEMMA 3: Let A be the adjacency matrix for a graph G. Then $(A^k)_{i,j}$ is precisely the number of distinct walks of length k between i and j.

We can restate our first two lemmas in terms of the adjacency matrix and thus characterize all of the problems under discussion in a form convenient for our algorithms. The diameter of a graph can be characterized as follows.

LEMMA 4: A graph G has diameter k iff
(1) for every i and j, either $(A^{k-1})_{i,j} > 0$ or $(A^k)_{i,j} > 0$, and
(2) for some i and j, $(A^{k-2})_{i,j} = 0$ and $(A^{k-1})_{i,j} = 0$.

The characterization for the radius is similar.

LEMMA 5: G has radius k iff
(1) for some i, for all j, $(A^{k-1})_{i,j} > 0$ or $(A^k)_{i,j} > 0$, and
(2) for all i, for some j, $(A^{k-2})_{i,j} = 0$ and $(A^{k-1})_{i,j} = 0$.

Notice that having found the diameter of G we can easily find the periphery according to the rule that

$$periphery(G) = \{i \mid \text{for some } j, \text{ both } A_{i,j}^{k-2} = 0 \text{ and } A_{i,j}^{k-1} = 0\}.$$

Here k is the computed diameter of G. If k is the computed radius of G we find the center according to the rule that

$$center(G) = \{i \mid \text{for all } j, \text{ either } (A^{k-1})_{i,j} > 0 \text{ or } (A^k)_{i,j} > 0\}.$$

All pairs of peripheral points can be calculated in a similar manner and a breadth-first search will easily find a geodesic between any such pair. These last two computations can be carried out in $O(n^2)$ operations given the appropriate matrices A^{k-2}, A^{k-1}, and A^k.

EXTREMAL DISTANCES

Our algorithms are designed to utilize any fast method for multiplying $n \times n$ matrices [6,8]. Our running times will have corresponding improvements should the upper bound on matrix multiplication be improved.

The basic idea is to precompute the power-of-two powers of the adjacency matrix and to then conduct a binary search to find the k which satisfies the criterion of lemma 4 or 5. Our algorithm is asymptotically faster than Floyd's because it uses a Strassen-like matrix multiplication routine.

We construct an algorithm to find the diameter of a graph given its adjacency matrix as input. The final value of k corresponds to the diameter of G as given by lemma 4. The matrix B is related to the kth and $k+1$st powers of the matrix A and is used to test the conditions of lemma 4. At each iteration one additional bit of accuracy is added to k, so that after $\lceil \log n \rceil$ iterations the exact value of k is available.

Throughout the algorithm we use the notation $X > 0$ to mean that the matrix X has all of its entries non-zero and we use the notation $X \ngtr 0$ to mean the opposite, that X has at least one zero entry.

```
integer procedure DIAMETER($A$, $n$)
  begin
    { $A$ is the adjacency matrix of a connected $n$-vertex graph $G$ }
    if $n \leqslant 1$ then return 0;
    if $I + A > 0$ then return 1;
    $M_0 := A$;
    for $p := 1$ to $\lceil \log n \rceil$ step 1 do
        $M_p := M_{p-1} \cdot M_{p-1}$;
    $k := 2$;
    $B := I + A$;
    for $p := \lceil \log n \rceil - 1$ to 0 step $-1$ do
        { $B = A^{k-2} + A^{k-1} \ngtr 0$, $B \cdot M_{p+1} > 0$, and
          $M_l = A^{2^l}$ for $0 \leqslant l \leqslant \lceil \log n \rceil$ }
        if $B \cdot M_p \ngtr 0$ then
            begin
                $k := k + 2^p$;
                $B := B \cdot M_p$
            end;
    { $A^{k-2} + A^{k-1} \ngtr 0$ and $A^{k-1} + A^k > 0$ }
    return $k$
  end
```

LEMMA 6: Let $M(n)$ be the number of operations required to multiply two $n \times n$ matrices. Algorithm DIAMETER computes the diameter of a connected n-vertex graph using $O(M(n) \log n)$ operations and $O(n^2 \log n)$ words of storage.

PROOF:

Correctness: If G has 0 or 1 vertices then trivially it has diameter 0 and the procedure obviously returns the correct value. We also notice that the only possible way that a graph can have a diameter of 1 is for the graph to be complete. Because the procedure explicitly checks for $I + A > 0$ this case is again handled correctly. Thus it remains only to verify that the procedure is correct when given a graph whose diameter is at least 2.

If this is true the procedure continues its initialization by building a table of the power-of-two powers of the matrix A. The 2^l power is stored in the matrix M_l for $0 \leqslant l \leqslant \lceil \log n \rceil$. The algorithm never uses any higher powers of the matrix because the diameter of an n-vertex graph is at most $n - 1$. The next two assignments initialize k and B. Thus whenever the program first reaches the second loop the assertion stated there is valid.

We next show that the assertion remains true each time the loop is executed if it is true before the loop is executed. Let us use B', p', and k' to denote the values of the corresponding variables after executing the entire loop and reserve B, p, and k to denote the values of the variables before executing any part of the loop. The loop consists of an if-statement, so there are two cases to consider.

In the case where $B \cdot M_p > 0$ we have no changes to the variables, except for the loop index p. Thus $B' = B$, $p' = p - 1$, and $k' = k$. Substituting these values into the invariant, we obtain

$$B' = B = A^{k-2} + A^{k-1} = A^{k'-2} + A^{k'-1} \ngtr 0,$$

$$B' \cdot M_{p'+1} \;=\; B \cdot M_p \;\not> 0.$$

Obviously none of the matrices M_l is changed by the loop, hence the assertion remains valid.
In the case where $B \cdot M_p \not> 0$ we have a similar argument. The M_l again are unchanged, but this time $B' = B \cdot M_p$, $p' = p - 1$, and $k = k + 2^p$. Substituting into the assertion, we find that

$$B' \;=\; B \cdot M_p \;=\; (A^{k-2} + A^{k-1}) \cdot A^{2^p} \;=\; A^{k'-2} + A^{k'-1} \not> 0,$$

$$B' \cdot M_{p'+1} \;=\; (B \cdot M_p) \cdot M_p \;=\; B \cdot M_{p+1} \;>\; 0.$$

Thus regardless of the path through the loop, the values of the variables after executing the loop satisfy the assertion whenever the values of the variables before executing the loop satisfy the assertion. The assertion is thus an invariant of the program.

From this it follows that the final assertion is also an invariant because the only way in which the return statement can be reached is when the second loop terminates with $p = -1$. (We assume that the variable is still defined after the loop exit only as a convenience in our proof.) This final invariant implies partial correctness of the program. Total correctness follows from the observation that termination is guaranteed because each loop is executed at most $\lceil \log n \rceil$ times.

Timing: As we have just remarked, each loop is executed at most $\lceil \log n \rceil$ times. The primary work within each loop consists of a single matrix multiplication. All of the other operations within the loops and within the rest of the program are dominated by these matrix multiplications. Hence the total number of operations is $O(M(n)\log n)$.

Storage: The largest space requirement is for the matrices M_l. Each of these has $O(n^2)$ entries and there are $\lceil \log n \rceil + 1$ of the matrices. Thus the total storage is $O(n^2 \log n)$. □

The procedure is easily modified to find the radius of a graph simply by looking for a single row or column of the matrix which contains no zeros, instead of demanding that the entire matrix contain no zeros. We have now shown the following.

THEOREM 7: We can find the diameter, radius, periphery, center, all pairs of peripheral points, or a geodesic of an n-vertex graph in $O(M(n)\log n)$ operations.

PROOF:

Using the procedure DIAMETER we can find the diameter or radius in the desired time. The periphery, center, and all pairs of peripheral points can be read out of the appropriate matrices A^{k-1} and A^k using the characterizations given in section 2. This requires at most a single scan over the two matrices which is only $O(n^2)$ additional work. Given a pair of peripheral points it is easy to find a connecting geodesic in $O(n^2)$ operations using a breadth-first search. □

If we use a Strassen-like method for matrix multiplication we improve both on Floyd's algorithm and on Fredman's. Pan's algorithm uses $O(n^{2.795})$ operations to multiply two $n \times n$ matrices. We can thus state the following numerical result.

COROLLARY 8: We can find all of the items of theorem 7 in $O(n^{2.795} \log n)$ operations.

It is worth noting that our approach is useless without a fast matrix multiplication scheme. If we use the naive $O(n^3)$ method our algorithms are actually worse than Floyd's algorithm, both in a practical sense and asymptotically.

The storage requirement for our algorithm is $O(n^2 \log n)$ because of the $\log n$ matrices which we precompute. We could store these using $O(n^2 \log n)$ bits because we really do not need the actual numerical values of the matrices, rather we are only interested in the zero or nonzero state of each entry. In an implementation this savings in storage might be worth the extra overhead required to store the bits in a packed form. The asymptotic time bounds would remain unchanged.

APPROXIMATE DISTANCES

In the previous section we discovered that extremal distances can be computed in less time than required by current algorithms for computing the entire distance matrix. If we are interested in computing all of the distance matrix, but will settle for having our answers correct only to within a factor of two, our algorithms also apply. We are essentially asking for the integer parts of the logarithms of the distances. The power-of-two powers of the adjacency matrix computed earlier provide sufficient information to determine all distances to within a factor of two using $O(M(n)\log n)$ operations.

We can actually obtain approximate distances correct to within any constant factor if we are willing to perform a little more work. For any $\epsilon > 0$ define $\phi = 1+\epsilon$. If we let

$$\hat{d}(i,j) = \min\{ \lfloor \phi^k \rfloor \mid (A^{\lfloor \phi^k \rfloor})_{i,j} > 0 \}$$

then $\hat{d}(i,j)$ is a good approximation to $d(i,j)$. This is formalized by the next lemma.

LEMMA 9: For all $i \neq j$, $\dfrac{|\hat{d}(i,j) - d(i,j)|}{d(i,j)} < \epsilon$.

PROOF:

Since $i \neq j$ we must have $d(i,j) > 0$. Choose the minimal k such that $\hat{d}(i,j) = \lfloor \phi^k \rfloor$. By definition,

$$\lfloor \phi^{k-1} \rfloor < d(i,j) \leqslant \lfloor \phi^k \rfloor = \hat{d}(i,j).$$

Because $d(i,j)$ is an integer we can ignore the floor function and simply write $\phi^{k-1} < d(i,j)$. Thus we find that

$$\hat{d}(i,j) = \lfloor \phi^k \rfloor \leqslant \phi^k < \phi \cdot d(i,j).$$

But $\phi = 1+\epsilon$ so when we subtract $d(i,j)$ from each side we are left with

$$\hat{d}(i,j) - d(i,j) < \epsilon \cdot d(i,j).$$

Because $\hat{d}(i,j)$ is no smaller than $d(i,j)$ we conlcude that

$$\frac{|\hat{d}(i,j) - d(i,j)|}{d(i,j)} < \epsilon$$

which is the desired result. □

To compute $\hat{d}(i,j)$ for all i and j we need only compute the sequence of matrix powers $A^{\lfloor \phi^0 \rfloor}$, $A^{\lfloor \phi^1 \rfloor}, \cdots, A^{\lfloor \phi^k \rfloor}, \cdots$, stopping when the nth power of A is reached. We then check to see when each (i,j)-entry first becomes nonzero in one of the matrices, recording the corresponding power as $\hat{d}(i,j)$.

THEOREM 10: Given any $\epsilon > 0$ there is an $O(M(n)(\log n)^2)$ algorithm (whose constant of proportionality depends upon ϵ) which computes approximate distances $\hat{d}(i,j)$ having a relative error no larger than ϵ.

PROOF:

We compute $\log_\phi n$ matrix powers. Each matrix can be computed using at most $2\log n$ matrix multiplications by inspecting the binary expansion of its exponent. These computations dominate the remaining work. If we note that $\log_\phi n$ is actually $\log n/\log \phi$ then an upper bound on the total number of operations is $O(M(n)(\log n)^2)$. In terms of ϵ the constant of proportionality varies as $1/\log(1+\epsilon)$ which is $1/\epsilon$ for sufficiently small ϵ.

The number ϕ is of course not an integer. We must keep enough of the low-order (fractional) bits to ensure that the truncated value $\lfloor \phi^k \rfloor$ is correct in its integer part. If we initially compute ϕ to $\log_\phi n$ fractional bits the computation stops when ϕ^k reaches n so the cumulative error will not propagate into the integer part of our calculation.

Technically we would require a different algorithm for each n because the actual data constant which represents ϕ depends upon n, but in practice we can always choose $\epsilon = 2^{-t}$ and eliminate this problem. □

COROLLARY 11: Given any $\epsilon > 0$ there is an $O(n^{2.795}(\log n)^2)$ algorithm which computes approximate distances having a relative error no larger than ϵ.

The storage requirement for this algorithm is clearly $O(n^2)$ because we have only a fixed number of matrices in use at any time. It is interesting to ask whether the second factor of $\log n$ is really necessary in theorem 10. Is it possible to calculate the $\lfloor \phi^k \rfloor$-th power of A from the $\lfloor \phi^{k-1} \rfloor$-th power of A using only a fixed number of matrix multiplications? We leave this as an open question.

For arbitrary values of ϕ we see no better method than examining the binary expansion of $\lfloor \phi^k \rfloor$ to obtain $A^{\lfloor \phi^k \rfloor}$, even if we have available all of the powers of A previously computed. (This method would, of course, require additional storage.) Perhaps there is an easier method.

ALL DISTANCES

If we want to compute all distances instead of just the extremal distances our techniques do not always work. If the graph is sparse we can take advantage of its sparseness by computing distances using a breadth-first search beginning at each vertex [5]. This will yield an $O(ne)$ algorithm which for $e < n^2$ will be better than Floyd's algorithm. The problem seems to be with dense graphs.

It would seem that with many edges a dense graph would be likely to have a small diameter. If the diameter is small then we can compute the matrices $A^0, A^1, A^2, \ldots, A^{diameter(G)}$ to find all distances in the graph, a method similar to that outlined in theorem 10. This would require at most $O(M(n) \cdot diameter(G))$ operations. For arbitrary graphs we could then use a hybrid approach, depending upon the density of edges, and perhaps achieve a uniformly fast algorithm.

But density alone does not insure a small diameter [10]. It is easy to exhibit graphs with almost $n^2/4$ edges whose diameter is n. If we want to employ our hybrid solution we need a more uniform condition than density.

Define a graph to be *pseudo-regular of degree* ρ iff the degree of each vertex is bounded above and below by

$$\rho/2 \leqslant degree(i) \leqslant 2\rho$$

so that all vertices have about ρ neighbors. Note that all regular graphs are pseudo-regular but that the converse is not true.

LEMMA 12: If G is pseudo-regular graph of degree ρ then $diameter(G) \leqslant 6n/\rho$.

PROOF:

Consider a geodesic of G whose vertices are $i_0, i_1, \cdots, i_{diameter(G)}$. Because a geodesic is a shortest path between its endpoints the neighbors of i_j and i_{j+3} must be distinct for every j. Thus the total number of vertices on the geodesic and adjacent to it must be at least $\lceil diameter(G)/3 \rceil \cdot \lceil \rho/2 \rceil$. This cannot exceed the number of vertices in the graph, hence we know that $diameter(G) \leqslant 6n/\rho$. □

Our hybrid algorithm will want to balance the cost of the breadth-first search for sparse psuedo-regular graphs against the cost of performing $diameter(G)$ matrix multiplications when the graph is dense. Ignoring constants and low-order terms we must solve the equation

$$n M(n)/\rho = n^2 \rho$$

for ρ, where the left-hand side is the cost for dense graphs using our matrix technique, and the right-hand side is the cost for sparse graphs using breadth-first search. The crossover occurs when

$$\rho = (M(n)/n)^{1/2}$$

at which point the number of operations is $O((n^3 M(n))^{1/2})$. We have shown the following result.

THEOREM 13: There is an $O(n^{2.898})$ algorithm for computing all distances in any pseudo-regular graph.

This algorithm achieves the same running time as does Johnson's algorithm [5] if the input graph is sparse, but our algorithm is superior if the number of edges exceeds $n^{1.898}$. Recall, however, that our algorithm is only designed to work well for pseudo-regular graphs whereas Johnson's algorithm works well for any graph.

It is interesting to note that if we use a naive matrix multiplication instead of Strassen's method we won't improve on Floyd's algorithm and in the unlikely event that $M(n) = O(n^2)$ we would match the $O(n^{2.5})$ decision tree complexity of the algorithm given by Fredman [3].

CONCLUSIONS

We have exploited the fact that it is unnecessary to know all distances within a graph when computing various distance statistics such as the diameter and radius of a graph. These same techniques have also been used to compute approximate distances between all pairs of vertices in a graph with unit cost edges. Finally, we have achieved an improved running time for the exact all pairs problem in graphs with unit cost edges under the assumption of pseudo-regularity.

We note without proof that if a graph is chordal [7] we can compute the exact distance matrix in $O(n^2)$ operations using the observation in [9] that all minimal separators are cliques. The graphs which demonstrate that high density does not guarantee small diameter are chordal. It is thus interesting that there is still an efficient algorithm for computing their distance matrices. The algorithm does not employ the methods introduced here so we will not discuss the details.

It would be nice to find other classes of graphs for which the all pairs problem can be solved efficiently. We suspect that distances are not as hard to compute for unit cost edges as for arbitrary costs and propose this as an area for further research.

ACKNOWLEDGEMENT

This work was supported in part by the National Research Council of Canada under grant A4307 and by the Office of Naval Research under grant N00014-75-6-0752.

REFERENCES

[1] J. A. Bondy and U. S. R. Murty, *Graph Theory With Applications*, MacMillan, London, 1976.

[2] R. W. Floyd, Algorithm 97: Shortest path, *Communications of the ACM* 5(6):345, June 1962.

[3] M. L. Fredman, New bounds on the complexity of the shortest path problem, *SIAM Journal On Computing* 5(1):83-89, March 1976.

[4] F. Harary, *Graph Theory*, Addison-Wesley, Reading, Massachusetts, 1969.

[5] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, *Journal of the ACM* **24**(1):1-13, January 1977.

[6] V. Pan, Strassen's algorithm is not optimal, to appear in *IEEE* 19th *Annual Symposium on Foundations of Computer Science*, 1978.

[7] D. J. Rose, R. E. Tarjan, and G. S. Lueker, Algorithmic aspects of vertex elimination on graphs, *SIAM Journal on Computing* 5(2):266-283, June 1976.

[8] V. Strassen, Gaussian elimination is not optimal, *Numerishe Mathematik* **13**:354-356, 1969.

[9] Y. Perl and Y. Shiloach, Finding two disjoint paths between two pairs of vertices, *Journal of the ACM* **25**(1):1-9, January 1978.

[10] V. G. Vizing, The number of edges in a graph of given radius, *Soviet Mathematics - Doklady* **8**(2):535-536, 1967.

A GENERALIZED ALGORITHM FOR CENTRALITY PROBLEMS ON TREES

ARNIE ROSENTHAL
The University of Michigan, Computer and Communication Sciences Dept.
Ann Arbor, Michigan 48109

ABSTRACT: A very general method is presented for rapidly analyzing tree networks to compute a measure of the "centrality" or "eccentricity" of all vertices in the tree. A new widely applicable way of presenting generalized algorithms is used. We consider a number of realistic cost criteria, some quite complicated to evaluate. Many of these problems have no published solution better than performing a separate traversal for each vertex whose eccentricity is calculated, but our method performs just two traversals, works for all these measures, and yields the eccentricities of all points in the tree. Natural sufficient conditions for the method to work are formally expressed.

Keywords: GENERALIZED ALGORITHM, FACILITY LOCATION, LINEAR TIME, TREE TRAVERSAL, CENTER, ECCENTRICITY, CENTROID, CENTRALIZED NETWORK, AXIOMATIC SPECIFICATION, UNINTERPRETED FUNCTION

1. Introduction

Many optimization problems arising in applications can be modelled as the finding of a vertex in a graph which is most central relative to a given cost structure (see Christofides [1], for example). For each cost structure there is a corresponding definition of the "eccentricity" of a vertex such that a "center" is a vertex with minimal "eccentricity". A large number of algorithms for finding centers, each designed for a single cost structure, have appeared in the literature [2, 3, 4, 5, 7, 8, 11].

We restrict ourselves to undirected trees, and consider the more general problem of computing the "eccentricity" or "cost" for all vertices in the tree. (This approach allows later inclusion of cost factors not included in the network model.) Admissible cost structures must satisfy the following rather weak condition: if one roots the tree at an arbitrary vertex x, it must be possible to traverse the tree bottom up, recursively computing information (called subtree weights) such that the eccentricity of x may easily be computed from the weights of subtrees incident to x.

To compute eccentricity or "total" cost for all vertices, we introduce a generalized "schematic" algorithm. The method for expressing the algorithm's generality merits attention. Formally, the local cost structure is specified by subroutines (i.e., uninterpreted function symbols). Conditions (i.e., axioms) are explicitly given such that for any admissible local cost functions (i.e., under any interpretation), the algorithm will be valid and will run in linear time. The user supplies subroutines which specify the cost structure by calculations to be performed at each vertex. Our algorithm handles the global tree traversal and cost calculations.

Given the cost structure of a tree, define the eccentricity of a vertex v in that tree (denoted simply ecc(v)) to be the cost of distributing to all the vertices of the tree from a center located at v.

The general problem: Given a tree and its cost structure, find the eccentricities of all vertices in the tree.

Previous work: In many of the applications below, attention has centered on computing ecc(root) in linear time for a complicated cost structure. To find all eccentricities, one must repeat the computation for each vertex, for a total of $O(|V|^2)$. Except for the simplest applications (iii and iv below), there is no previous algorithm which handles even a single cost structure in linear time. Running time for previous state of the art algorithms are given for each example application.

Our algorithm takes linear time, but works for many cost functions. A further benefit is that the tree traversal is conceptually isolated from what may be a very intricate cost structure.

2. Terminology. Let T be an undirected tree graph with vertex set V and edge set E. As a matter of convenience for defining traversals, we shall consider it to be rooted at some arbitrary vertex r_o.

The following notation will be used:

$|S|$ = number of elements of set S.

Father (v) = The vertex immediately following v on the path from v to r_o. (By convention r_o does not have a father.)

Adj(v) = {vertices adjacent to v}

Sons (v) = Adj(v) - {Father (v)} .

degree(v) = |Adj(v)|.

ecc(v) = the total cost of the network if centered at v.

(u,v) = oriented edge from vertex u to vertex v (u and v being adjacent).

T(u,v) = {all vertices and edges reachable from v in T - {(u,v)}. (See figure 1.)

In each example application below, each subtree is given an associated weight denoted wt(u,v) from which the eccentricities are computed. The weight wt(u,v) of a subtree $T_{(u,v)}$ is a sufficient summary of the subtree -- sufficient in the sense that:

i) wt(u,v) contains all the information about $T_{(u,v)}$ relevant to computing ecc(v), and

ii) If $T_{(w,y)} \supseteq T_{(u,v)}$, wt(u,v) contains all information about $T_{(u,v)}$ relevant to computing wt(w,y).

The crucial idea is that for "reasonable" cost functions, it is possible to obtain such a compact representation of any subtree. Intuitively, this representation expresses the contribution of the subtree to the eccentricity of any point outside the subtree.

WTS(v) ≡ the set of weights wt(v,x) such that $x \in$ Adj(v)

OTHER_WTS(u,v) ≡ WTS(v) - {wt(v,u)}

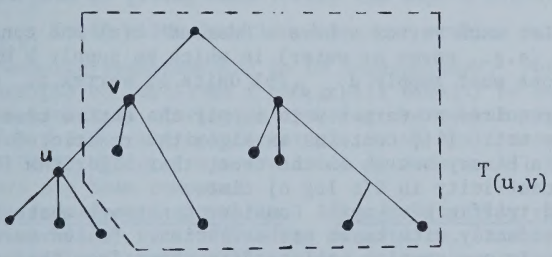


Figure 1 - The oriented edge (u,v) and its subtree $T_{(u,v)}$ (in dotted box)

3. Applications. Before presenting the algorithm, we list below some optimization problems, together with the corresponding eccentricity definitions. In all but one example, each vertex x is to receive a volume c(x) of traffic (e.g., messages) from a central facility. Each message has only one recipient. The traffic (denoted $t_{(x,y)}$) carried by an edge (x,y) depends on the placement of the central facility; all vertices from the subtree

[$T_{(x,y)}$ or $T_{(y,x)}$] *not* containing the center have their traffic routed over (x,y). The value of ecc(v) will be determined by those (center-dependent) edge traffics.

(i) *Minimizing a sum of traffic-dependent edge costs.* We wish to minimize total message costs. Suppose it costs $d_{(x,y)}(t)$ to send t units of traffic across edge (x,y). Let ecc(v) be the sum of all these traffic-dependent) edge costs. Previous state of the art: References [2,17] consider this problem in the special case where all d(.) are linear. Details of their implementation are not given, but the method seems to require at least O(|V|) for each eccentricity computed. (General graphs as well as trees are considered in these papers.)

(ii) *Telecommunications amplifier problem.* In some telecommunication systems (e.g., cable television) a signal can travel only a limited distance without being amplified. We wish to locate the central signal distribution point so as to minimize the number of amplifiers required. ecc(v) is the number of amplifiers required if the central distribution point is v. Previous state of the art for this problem has been θ(|V|) for each eccentricity computed.

(iii) *Minimize maximum congestion.* We wish to minimize the maximum traffic carried by any single edge. If v is the center, the maximum traffic will be on one of the edges (v,x) out of v. Define $ecc(x) = \max \{t_{(v,x)} | x \in adj(v)\}$. When each vertex receives 1 unit of traffic, the example reduces to the problem of finding the "centroid" of a tree[6]. Previous state of the art for this problem has been Θ(|V|) for finding the centroid.

(iv) *Minimax distance location problem.* Consider the problem of locating an emergency facility or pizza parlor for communities connected by a tree of roads. We wish to minimize the longest distance a patient or pizza must travel.

Define the eccentricity of a vertex to be the length of the longest path from the vertex to any other vertex.

The simple case in which all the edge distances are the same is the problem of finding the "center" of the tree as defined in graph theory [6].

In [2], an additional vertex-dependent time delay is added in at each vertex. This additional delay is easily handled. Once again, the previous state of the art for this problem has been Θ(|V|) for finding the best distribution point.

(v) *Lossy network.* Let each vertex x have a "demand" c(x) and consider a distribution network (e.g., power or water) in which to supply b units at end y of edge (e,y) one must supply $d_{(x.y)}(b)$ units at vertex x. ecc(v) is the number of units required at vertex v to supply the entire tree from v. Previous state of the art: [11] contains an algorithm restricted to linear d(.). By performing a binary search on the tree, that algorithm finds a vertex of minimum eccentricity in O(n log n) time.

(vi) *Minimum expected traffic blockage.* Consider a network whose vertices and edges fail independently with known probabilities. Failed vertices block communication. In one version, all traffic comes from the center, and ecc(v) ≡ expected number of vertex pairs disconnected by failures. Previous state of the art: [8] gives an algorithm for each version of the problem, in which O(|V|) time is required for each eccentricity computed.

4. *Generalized algorithms.* We want to separate the main algorithm from the details of the cost structure. For each application we will define subtree "weights" which will serve as sufficient summaries of the subtrees. For example, in the centroid example (iii), the weight of a subtree is simply the amount of traffic routed toward the subtree. The appendix contains weight definitions for the other applications.

While traversing the tree, the generalized algorithm will call subroutines (i.e., uninterpreted functions) f and g to compute weights and eccentricities. g takes the weights of all subtrees incident to a vertex v, and computes ecc(v). f takes as argument an edge (u,v) and weights of subtrees $\{T_{(v,z)} | z \neq u\}$ (i.e., OTHER_WTS(u,v)) and computes the weight wt(u,v) of their union $T_{(u,v)}$. The task of the tree traversal algorithm is to call f and g with the proper arguments when the relevant weights are known.

The manipulations performed by f and g are not complicated. In all the applications mentioned there must be algorithms which traverse a rooted tree bottom-up to compute ecc(root). At each v≠root, these algorithms use a function analoagous to f to compute the weight. An analogue of g is used to compute ecc(root). Our contribution is to abstract this manipulation, and then observe that a single downward traversal from an arbitrarily chosen root can now provide all vertices' eccentricity.

Axioms 1 and 2 below are conditions on the cost structure. They assert that the necessary functions f and g for defining and manipulating weights do indeed exist. The axioms give a rigorous meaning to the statement that weights must effectively summarize subtrees. Note that by axiom 2, the weight of a subtree consisting of a single leaf may always be computed by a function with zero input arguments.

Axioms. For every vertex v and any oriented edge (u,v), it is possible to define weights and functions g_v, $f_{u,v}$ such that:

Axiom 1. $\quad ecc(v) = g_v(WTS(v))$

Axiom 2. $\quad wt(u,v) = f_{u,v}(OTHER\_WTS(u,v))$

In order to perform a meaningful analysis of running time, it will be necessary to specify running times for f and g. It will not be sufficient to count function calls, since some invocations take much more time than others.

Axiom 1T. $\quad g_v(.)$ may be computed in time O(deg(v)).

Axiom 2T. $\quad f_{(u,v)}(.)$ may be computed in time O(deg(v)).

Axioms 1T and 2T are designed so that the overall algorithm will take linear time on trees without high-degree vertices. These axioms also rule out trivially defining the weight of a subtree as the subtree itself. To satisfy 1T and 2T, weights will need to be succinct summaries.

The purpose of giving each vertex and edge a different function g_v (resp., $f_{u,v)}$ is to allow location dependent problem data (e.g., edge lengths, resource requirements) to be incorporated. For notational simplicity, subscripts on functions f and g will usually be omitted.

The mechanics of storing and passing the vector arguments to the subroutines for f and g are uninteresting, so we shall assume that the vectors WTS(v) and OTHER_WTS(u,v) are available as soon as all their component weights have all been computed.

5. A simple (nonlinear) algorithm. After arbitrarily choosing a root r_0, it will be necessary to traverse the tree twice, once upward, once downward. In the first traversal, before a vertex is processed, all of its sons must be processed. This will be called a "bottom-up traversal". In the second traversal, a vertex is always processed before any of its sons. This is called a top-down traversal. The conventional postorder [preorder] traversal is a satisfactory bottom-up [top-down] traversal.

The first algorithm for computing eccentricities may now be presented. The use of functions f and g directly exploits the axioms.

Algorithm 1

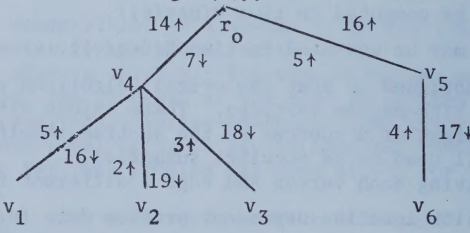
```
      PROCEDURE central_points;
      BEGIN
(1)      v:=first_vertex_of_bottom_up_traversal;
```

```
(2)     WHILE v ≠ r_o
            BEGIN
(3)             wt(father(v),v):=f(OTHER_WTS(father(v),v));
(4)             v:=bottom_up_successor(v)
            END;
(5)     v:r_o;      //SECOND TRAVERSAL//
(6)     ecc(r_o):=g(WTS(r_o))
    REPEAT
(7)     For each child w of v
            BEGIN
(8)             wt(w,v) :=f(OTHER_WTS(w,v));
(9)             ecc(w) :=g(WTS(w))
            END;
(10)    v :=TOP_DOWN_SUCCESSOR(v);  //(non-leaves) top down//
(11) UNTIL ALL_NON_LEAF_VERTICES_HAVE_BEEN_VISITED
    END;
```

If we represent the tree T in a drawing with r_o on top, then step 3 of Algorithm 1 computes "downward-looking" subtree weights, whereas step 8 computes the "upward-looking" ones (see figure 2).

As an example of the operation of the CENTRAL_POINTS algorithm, consider the centroid problem for the tree depicted in figure 2. Each value wt (x,y) is represented on the edge (x,y) close to the vertex x.

$c(v_1) = 5$
$c(v_2) = 2$
$c(v_3) = 3$
$c(v_4) = 4$
$c(v_5) = 1$
$c(v_6) = 4$
$c(r_o) = 2$



$ecc(v_1) = 16$
$ecc(v_2) = 19$
$ecc(v_3) = 18$
$ecc(v_4) = 7$
$ecc(v_5) = 16$
$ecc(v_6) = 17$
$ecc(r_o) = 14$

Figure 2.

Tree rooted at r_o. Weights computed after the first (↑) and second (↓) loops of algorithm 1.

The f function in this case is

$$wt(u,v) := c(v) + \sum_{\substack{x \text{ adj } v \\ x \neq u}} wt(v,x)$$

Thus wt (r_o, v_4) is computed as 4 + (5 + 2 + 3)

6. Analysis of Algorithm 1. Step 8 is executed O(degree(v)) times for each non-leaf vertex v, and each execution takes time O(degree(v)), according to Axiom 2T. Therefore, step 8 requires time $O(\sum_{v \in V-\{leaves\}} (deg(v))^2)$ to be executed for all v. Thus if there is a vertex of degree $O(|V|)$, then algorithm 1 takes time $O(|V|^2)$ to execute. However, if the maximum degree of a vertex is bounded by a constant, then algorithm 1 executes in time $O(|V|)$.

7. A Linear Time Algorithm. As the reader will notice, wt is retrieved from memory when needed, not recalculated. To obtain an algorithm that executes in linear time on all trees, we need only to introduce additional memory to

store other intermediate calculations. Let SUM(v) represent an additional value to be associated with vertex v "SUM" will hold a "sum" of all the subtrees from v, and for $x \in adj(v)$, wt(x,v) will be computable by "subtracting" wt(v,x) from SUM(v). Rather surprisingly, every application considered, it is possible to define such a "subtraction" operation (though occasionally SUM must be a 2-element vector).

The simplest example of a SUM function is found in the centroid problem (iii). Recall $wt(u,v) = \sum_{x \in T_{(u,v)}} c(x)$. If we define SUM by

$$SUM(v) = \sum_{x \in adj(v)} wt(v,x),$$ then $wt(u,v) = SUM(v) - wt(v,u) + c(v)$.

Our second algorithm modifies the downward traversal to incorporate the trick with "SUM" and eliminate the time-consuming step 8. The upward traversal and axioms 1 and 2 will be unchanged. Axioms 1' - 4' specify the conditions for the second generalized algorithm.

For all vertices v and edges (u,v), it is possible to define weights and functions g_v, $f_{u,v}$, $f2_v$, $f3_{u,v}$ such that:

Axiom 1' $ecc(v) = g_v(WTS(v))$

Axiom 2' $wt(u,v) = f_{u,v}(OTHER\_WTS\ \hat{}\,(u,v))$

Axiom 3' $SUM(v) = f_{2_v}(WTS(v))$

Axiom 4' $wt(v,u) = f_{3_{u,v}}(SUM(u), wt(u,v))$

Again, for notational simplicity, the subscripts u and v for functions will be omitted.

We shall specify the conditions on timing under which the algorithm will be analyzed.

Axiom 1T' g_v takes time O(deg(v))

Axiom 2T' $f_{u,v}$ takes time O(deg(v))

Axiom 3T' f_{2_v} takes time O(deg(v))

Axiom 4T' $f_{3_{u,v}}$ takes time O(1)

Algorithm 2

```
PROCEDURE central_points;
BEGIN
(1)   v:=first_vertex_of_bottom_up_traversal;
(2)   WHILE v ≠ $r_o$
        BEGIN
(3)       wt(father(v),v):=f[OTHER_WTS(father(v),v))];
(4)       v:=bottom_up_successor(v)
        END;
(5)   v:=$r_o$;
(6)   SUM($r_o$):=$f_2$(WTS($r_o$));
(7)   ecc($r_o$):=g(WTS($r_o$));
      REPEAT
(8)       For each child w of v
              BEGIN
(9)               wt(w,v) :=$f_3$[SUM(v), wt(v,w)];
(10)              SUM(w) := $f_2$(WTS(w));
```

```
(11)        ecc(w) :=g(WTS(w))
        END;
(12)      v := TOP_DOWN_SUCCESSOR(v)  //interate thru internal vertices
                                           (nonleaves)  top down//
(13)  UNTIL ALL_NON_LEAF_VERTICES_HAVE BEEN_VISITED
    END;
```

8. Analysis of Algorithm 2. The first traversal has not been altered; it still takes time $O(|V|)$. Axiom 3T' guarantees that steps 6 and 10 will account for time of $\sum_{v \in V} O(\text{degree}(v))$ which is $O(|V|)$. From Axiom 4T' we know that each execution of step 9 takes a constant amount of time. Since step 9 is executed $|V| - 1$ times it takes $O(|V|)$ time to execute.

Thus algorithm 2 executes in time $O(|V|)$. To find the vertex with minimal eccentricity it suffices to carry out a linear search of the computed eccentricities.

9. Specifying general conditions. In describing the conditions under which the algorithms work, we became aware of three factors governing the choice of a set of axioms. The factors are useful generality, ease of verifying, and elegance. Not surprisingly, they can conflict. For instance, the most general axiom under which an algorithm in linear time must be one which says "The algorithm works correctly, and in linear time". Any other axiom will be more restrictive. The objection to this axiom is that it is a disaster on the second criterion; verifying that the axiom holds requires proving from scratch the algorithm's correctness and running time.

The question of elegance arose when the conditions for algorithm 2 were specified. The reader may have noticed that Axiom 2' need not apply when $u \neq \text{father}(v)$, since different functions were used on the walk down the tree.

Thus, the condition could be weakened to: It is possible to root the tree at some r_o (and thereby define a 'father' relation) such that for any $v \neq r_o$ there exists a function f_v such that wt(father(v),v) =

f_v(OTHER_WTS(father(v,v)).

The problem with this reformulation is that in pursuing the unattainable goal of complete generality, it has traded away too much elegance. We have found no facility location problem which requires the added generality, so the trade seems inadvisable here. One might briefly note the possibility of weakening the condition at the root.

In the problem considered here, it was necessary to explicitly state the assumptions about running times of various functions. This approach will often be more advantageous than simply counting invocations in situations where different invocations require different amounts of time.

REFERENCES

1. Christofides, N., Graph Theory, an Algorithmic Approach. Academic Press, New York, 1975, Ch. 5-6.

2. Goldman, A.J., Minimax location of a facility in a network. Transp. Science, 6, 4(Nov. 1972), 407-418.

3. Hakimi, S.L., Optimum locations of switching centers and the absolute centers and medians of a graph. Operations Research, 12, 3 (May-June 1964), 450-459.

4. Halfin, S., On finding the absolute and vertex centers of a tree with distances. Transp. Science, 8, 1 (Feb., 1974), 75-77.

5. Handler, G.Y., Minimax location of a facility in an undirected tree graph. Transp. Science, 7, 3 (Aug. 1973), 287-293.

6. Harary, F., Graph Theory. Addison-Wesley, Reading, 1969, Ch. 2-4.

7. Hua, Lo-keng, et al., Application of mathematical methods to wheat harvesting. Mathematica Sinica, 11, 1 (1961), 77-91.

8. Kershenbaum, A. and Van Slyke, R., Recursive analysis of network reliability. Networks 3, 1973, 81-94.

9. Knuth, D.E. Fundamental Algorithms. Addison-Wesley, Reading, Mass., 1968, Ch. 2.

10. Rosenthal, A., Optimal algorithms for parameter variation, to appear.

11. Shekel, J., Optimal source location to feed a lossy distribution tree, IEEE Trans. Circuit Theory, 20, 3(May 1973), 246-250.

APPENDIX

Below we describe weights used by algorithm 1 for the examples in the applications section. Formulas for the functions f and g are given. In some examples, each weight is itself a vector, with components denoted by subscripts (i.e., $wt(u,v)$, $wt_2(u,v)$, etc.). Space constraints force the omission of formulas for algorithm 2.

The functions MAX and TOTAL with vector arguments will yield the largest component of the vector and the sum of the components of the vector, respectively. For example,

$$\mathrm{MAX}(\mathrm{WTS}_1(v)) = \mathrm{Max}\{wt_1(v,z) \mid z \varepsilon \mathrm{Adj}(v)\}$$

i) Minimizing a sum of traffic-dependent edge costs.

Algorithm 1

The weight of a subtree $T_{(u,v)}$ is a two component vector. $wt_1(u,v)$ is the total traffic in the subtree. $wt_2(u,v)$ is the cost of transporting the traffic from u to the subtree's vertices.

(f). $wt_1(u,v) := \mathrm{TOTAL}(\mathrm{OTHER\_WTS}_1(u,v)) + c(v)$

$wt_2(u,v) := \mathrm{TOTAL}(\mathrm{OTHER\_WT}_2(u,v)) + d_{(u,v)}(wt_1(u,v))$

(g). $ecc(v) := \mathrm{TOTAL}(\mathrm{WTS}_2(v))$

(ii) Minimizing amplifiers in a telecommunications system

The telecommunications problem was not fully specified, because it has many variants (all variants we have considered seem to obey our axioms). For the sake of example, suppose:

a) amplifiers may be placed anyplace on an edge
b) the distance downstream from an amplifier (or the distribution point) to the next amplifier (or end of line) cannot exceed a fixed length L.
c) there is no penalty for splitting a signal

Algorithm 1

The weight of a subtree $T_{(u,v)}$ must be a vector with two components. $wt_1(u,v)$ is the longest path from u to an amplifier in edge (u,v) or in subtree $T_{(u,v)}$. $wt_2(u,v)$ is the number of amplifiers in edge (u,v) and subtree $T_{(u,v)}$.

(f). $wt_1(u,v) := [\mathrm{MAX}(\mathrm{OTHER\_WTS}_1(u,v)) + d(u,v)] \bmod L$

$wt_2(u,v) := \mathrm{TOTAL}(\mathrm{OTHER\_WTS}_2(u,v)) + \lfloor [\mathrm{MAX}(\mathrm{OTHER\_WTS}_1(u,v)) + d(u,v)]/L \rfloor$

(g). $ecc(v) := \mathrm{TOTAL}(\mathrm{WTS}_2(v))$

(iii) Centroid problem (with associated demand c(v))

Algorithm 1

The weight of a subtree is the total demand of the vertices in the subtree.

(f). $wt(u,v) := \mathrm{TOTAL}(\mathrm{OTHER\_WTS}(u,v)) + c(v)$

(g). $ecc(v) := \mathrm{MAX}(\mathrm{WTS}(v))$

(iv) Minimax distance location problem

Algorithm 1

The weight of a subtree $T_{(u,v)}$ is the maximum distance from u to any vertex z in the subtree. Note that the path from u to z must have v as the first vertex after u.

(f). $wt(u,v) := \mathrm{MAX}(\mathrm{OTHER\_WTS}(u,v)) + d(u,v)$

(g). $ecc(v) := \mathrm{MAX}(\mathrm{WTS}(v))$

v) Lossy Network

Algorithm 1

The weight of a subtree $T_{(u,v)}$ is the number of units needed at u to supply the vertices in the subtree.

(f). $wt(u,v) := d_{(u,v)}$ [TOTAL(OTHER_WTS(u,v)) + (v)]

(g). ecc(v) := TOTAL(WTS(v))

vi) Minimizing expected blocked traffic

Algorithm 1

The weight of a subtree $T_{(u,v)}$ has two parts. $wt_2(u,v)$ is the expected block traffic within the subtree. c(v) is the traffic normally from v and p(v) is the probability that v will fail.) $wt_1(u,v)$ is the normal traffic from the subtree of u.

(f). $wt_1(u,v)$:=TOTAL($OTHER\_WTS_1(u,v)$) + c(v)

$wt_2(u,v)$:= (1 - p(v)) * TOTAL($OTHER\_WTS_2(u,v)$) + p(v) * $wt_1(u,v)$

(g). ecc(v) := TOTAL(WTS(v))

ORTHOGONAL PACKINGS IN TWO DIMENSIONS

BRENDA S. BAKER, University of California, Berkeley, CA 94720
E. G. COFFMAN, JR., University of California, Santa Barbara, CA 93106
RONALD L. RIVEST, Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

We consider problems of packing an arbitrary collection of rectangular pieces into an open-ended, rectangular bin so as to minimize the height achieved by any piece. This problem has numerous applications in Operations Research and studies of computer operation. We devise efficient approximation algorithms, study their limitations, and derive worst-case bounds on the performance of the packings they produce.

I. INTRODUCTION

Efficiently packing sets of rectangular figures into a given rectangular area is a problem with widespread application in Operations Research. Thus, one is inclined to attribute the scarcity of results on this problem, and others of its type, to inherent difficulty rather than to lack of importance. Motivated by the intractability of these problems, we define and analyze certain approximation algorithms. These algorithms are natural in the sense that they would probably be among the first to occur to anyone wishing to design simple, fast procedures for determining easily computed packings. The analysis of these algorithms leads to bounds on the performance of approximate packings relative to the best achievable.

In the remainder of this section we define the model to be studied and introduce notation. At that point we examine in more detail the applications which are served by the model, and we review the literature bearing on this and similar models. In section II the main results of the paper are presented and proved. Concluding remarks and a discussion of open problems are given in section III.

As illustrated in Fig. 1, we consider an "open-ended" rectangle, R, of width w and a collection of rectangles, also called pieces, organized into a list $L = (p_1, p_2, \ldots, p_n)$. Each piece is defined by an ordered pair $p_i = (x_i, y_i)$, $1 \leq i \leq n$, corresponding to the horizontal (x_i) and vertical (y_i) dimensions of the rectangle.

We are concerned with the packing or assignment of the pieces in L into R so as to minimize the height, h, of the packing; i.e., the maximum height, measured from the bottom edge of R, of the space occupied by any piece in the packing (see Fig. 1). In addition to the implicit requirement that the spaces occupied by distinct pieces be disjoint, we restrict attention to packings that are orthogonal and oriented. An orthogonal packing is one in which every edge of every rectangle is parallel to either the bottom edge or the vertical edges of R. An orthogonal packing is also oriented if the rectangles are regarded strictly as ordered pairs; i.e., a rectangle (x_i, y_i) must be packed in such a way that the edges of length x_i are parallel to the bottom edge of R. Thus, rotations of 90° (which preserve orthogonality) are not allowed.

Returning to applications we see that our model applies to industrial or commercial situations in which objects are to be packed on floors, shelves, truck beds, etc. where concern is limited to the objects in two prespecified dimensions. Another important application concerns systems containing a shared resource. A prime example is the main memory resource in multiprogrammed computer systems. In such systems a number of tasks compete for a resource which they can share, but only within the limit provided by the total amount of resource available.

This application of the model was defined almost 20 years ago by E. F. Codd [1] in a study of multiprogramming systems. More recently, Garey and Graham [2] considered a related problem oriented to multiprocessor systems. In their study arbitrary numbers of processors and additional resources were considered. The analysis focused on worst-case bounds on the ratios of schedule-lengths (packing heights) for arbitrary lists; approximation algorithms were not considered. Moreover, the model of resources is basically different: Whenever an amount of the resource is available, no matter how it is configured, it can be used to satisfy any demand no greater than this amount; i.e., fragmentation of the resource is not a consideration.

Little else appears to have been published which bears on the packing problem we have defined. Erdös and Graham [3] have shown that orthogonal packings of squares into rectangles are not always optimum; i.e., there exist examples for which all orthogonal packings have greater height than the minimum achievable by exploiting the ability to rotate the squares. Based on earlier work of Meir and Moser [6], Kleitman and Krieger have considered the problem of finding a smallest rectangle into which a collection of squares can be packed [4,5]. Specifically, they prove that a $\sqrt{2} \times 2/\sqrt{3}$ rectangle is always sufficient to pack a set of squares whose cumulative area is unity, and that no rectangle of smaller area can have this property.

Even when $x_i = x_j$ for all i and j our packing problem is intractable; indeed, it can be shown that it becomes the NP-complete make-span minimization problem [7]. Hence, we are moved to consider fast heuristics and how closely the packings they produce approach optimum packings. For this purpose we define the following class of packing algorithms, to be called bottom-up left-justified (or simply BL) algorithms. (Recall that pieces must be packed so as to preserve oriented, orthogonal packings.) Each such algorithm packs the pieces one at a time as they are drawn in sequence from the list L. When a piece is packed into R it is first placed into the lowest possible location, and then it is left-justified at this vertical position in R. In the sequel R will also be referred to as a bin.

Figure 1 shows a BL packing. Note that from a combinatorial point of view our problem remains essentially unchanged if we replace left-justification by right-justification and consider BR packings instead. Note also that two BL algorithms differ only in the ordering of L.

II. PERFORMANCE BOUNDS FOR BL PACKINGS

We shall see that the basic BL algorithm, using a poorly ordered list L, can perform arbitrarily badly relative to an optimization algorithm. Thus, it is natural to inquire about the improvement possible by ordering L on the basis of some simple measure of piece size. Some obvious orderings to consider are increasing height, decreasing height, increasing width, and decreasing width. With the proper ordering the improvement can indeed be striking, as we shall see. However, with a badly chosen ordering, we can be just as poorly off as before. In particular, this will be true if we order pieces by increasing width.

As a matter of notation let $h_{BL}(L)$ and $h_{OPT}(L)$ denote the respective heights of a BL and optimum packing of L. The list L will normally be clear by context and hence omitted in this notation.

Theorem 1. For any $M > 0$, there exists a list of pieces ordered by increasing width such that $h_{BL}/h_{OPT} > M$.

Proof: We shall define a class of lists which proves this result. First, let $k \geq 2$ be given and define $r_i = \max\{m | i \equiv 0 \bmod k^m\}$, $i > 0$. Thus, $r_i = 1$ if i is a multiple of 4 but not 16, $r_i = 2$ if i is a multiple of 16 but not 64, $r_i = 3$ if i is a multiple of 64 but not 256, etc.; $r_i = 0$ if i is not a multiple of 4.

The rectangles defined below are specified in the order they are packed. Let $s = k^{k-1}$. From left-to-right along the bottom row we pack $w = k^k$ unit-width pieces, the i^{th} of which has height $1 - r_i\varepsilon$, where ε is much smaller than 1. The remaining rectangles all have unit height but widths in the order given by

$$\begin{array}{rl} s \text{ of width } 1 & \text{, (the second row)} \\ s/k \text{ of width } k & \text{, (the third row)} \\ s/k^2 \text{ of width } k^2 & \text{, (the fourth row)} \\ \vdots & \\ s/k^{k-1} = 1 \text{ of width } k^{k-1} & \text{, (the (k+1)-st row)} \end{array}$$

Since $r_i = 0$ for i not a multiple of k, one obtains the "notching" structure illustrated in the first row of Fig. 2; i.e., every k^{th} piece is lower than the intervening $k - 1$ pieces of unit height. Thus, the $s = k^{k-1}$ unit squares of the second row are placed on top of every k^{th} piece of the first row. Now every k^{th} piece of the second row corresponds to every $(k^2)^{th}$ piece of the first row and reaches a height at most $2 - 2\varepsilon$ which is less than that of the intervening $k - 1$ pieces of the second row, all at height $2 - \varepsilon$. Thus, the third row left-justifies pieces of width k over each of the lower pieces of the second row. Note that a width exceeding $k - 1$ is necessary so that the piece width exceeds the width of spaces in the second row.

A similar pattern applies to the heights reached in the third row: The height reached by every k^{th} piece is determined by that of every $(k^3)^{th}$ piece of the first row. Thus, in the third row every k^{th} piece achieves a height at most $3 - 3\varepsilon$ while the remaining pieces all achieve a height of $3 - 2\varepsilon$. It follows that the fourth row left-justifies pieces of width k^2 over each of the lower pieces of the third row. In general, then, the spaces in the $(j+1)^{st}$ row $(j \geq 1)$ all have width $k^j(k-1)$, the s/k^j pieces have width k^j, and every k^{th} piece reaches a height less than the others. Thus, when the next s/k^{j+1} pieces of width k^{j+1} are added, they are placed on top of every k^{th} piece in the $(j+1)^{st}$ row and each abuts the piece to the left in the $(j+1)^{st}$ row. Overall, the pattern looks like Fig. 2, drawn for $k = 4$. (Distinctions of $O(\varepsilon)$ are not all represented, because of scaling.)

Since there are k+1 rows, we see that $h_{BL} = k+1 - O(k\varepsilon)$. But a different packing can be found which packs rows 2 through k+1 into one row of height 1. (Note that the sum of widths of all pieces in rows 2 through k+1 is $k^{k-1} \times 1 + k^{k-2} \times k + \ldots + 1 \times k^{k-1} = k^k = w$.) Therefore, an optimum packing has a height not exceeding 2. Hence, we obtain a BL packing at least k/2 times higher than an optimum packing. Since k is arbitrary, the result follows. □

A dramatic improvement in the performance of BL packings is obtained when the list of rectangles is ordered by decreasing width. In fact, the ratio of BL to optimum packing height is guaranteed to be no worse than 3 when L is in decreasing order by width. For the case of squares, where decreasing width is equivalent to decreasing height, the bound is further reduced to 2. First, we shall show that the bounds of 3 and 2 can be approached as closely as desired; thus, these bounds are best possible.

Theorem 2. For any $\delta > 0$ there exists a list L of rectangles ordered by decreasing width such that the BL packing gives a height h_{BL} for which

$$\frac{h_{BL}}{h_{OPT}} > 3 - \delta \qquad (1)$$

If the pieces are restricted to squares then an L can be found such that for any $\delta > 0$

$$\frac{h_{BL}}{h_{OPT}} > 2 - \delta \qquad (2)$$

Proof: We shall prove the second result first, since the first result is but a slight modification.

The list proving (2) corresponds to the "checkerboard" packing in Fig. 3. The pieces are all either unit squares or approximately 2×2 squares. In particular, the larger squares are disposed on the bottom of the bin with the dimensions stepping down by ε from piece to piece. Hence the assignment of pieces in the second row must be made from right to left according to the bottom-up rule. Since two unit squares exceed the dimension of any larger square and squares are left-justified, only one unit square is placed on each large square. Except for the first and last pieces, this type of assignment repeats on the second row since the "holes" in the second row all have width less than 1 and squares to the right are lower than squares to the left. In general, the i^{th} row of unit squares alternates holes and pieces except for at most the initial and final $i - 1$ pieces of the row. Note that an optimum packing can be found which, except for possibly the last row, is within $O(\varepsilon)$ of being fully occupied.

The edge effects inhibiting the waste of half the space in the BL packing consist of

1. The row of larger pieces on the bottom
2. The triangular shaped solidly packed collections of squares on the left and right of the packing.

Holding piece sizes constant, the influence of the first edge effect is reduced by increasing the height of the packing, while the second is attenuated by widening the bin. Let k be two greater than the number of rows of unit squares. If the width of the bin is selected to be k^2, then the area of the bottom row and side edge effects is $O(k^2)$. Thus, ignoring $O(\varepsilon)$ terms, we can find a list such that

$$\frac{h_{BL}}{h_{OPT}} = \frac{k^3}{k^3/2 + O(k^2)}$$

In the limit $k \to \infty$, we have the bound of 2.

For the case of rectangles, it is only necessary to augment the list for Fig. 3 by adding as a new, last piece a rectangle of unit width and a height which equals the height of the optimum packing corresponding to the new list. Omitting the details, the BL packing will correspond to Fig. 3 with the new piece placed on top. It is easy to verify that the height ratio can now be made to approach 3 as closely as desired. □

Theorem 3. Let L be a list of rectangles ordered by decreasing widths. Then

$$\frac{h_{BL}}{h_{OPT}} \le 3 \tag{3}$$

This bound is best possible in the sense of Theorem 2.

Proof: Let h* denote the height of the lower edge of a tallest piece whose upper edge is at height h_{BL}. If y denotes the height of this piece, then $h_{BL} = y + h^*$. Let A denote the region of the bin up to height h*.

Suppose we can show that A is at least half occupied. Then we have $h_{OPT} \ge \max\{y, h^*/2\}$ and hence, $y > \frac{h^*}{2}$ implies

$$\frac{h_{BL}}{h_{OPT}} \le \frac{y+h^*}{y} < \frac{y+2y}{y} = 3$$

and if $y \le h^*/2$, we have

$$\frac{h_{BL}}{h_{OPT}} \le \frac{h^*/2 + h^*}{h^*/2} = 3$$

The result will thus be proved. It remains to show that A is at least half occupied.

Any horizontal cut or line through A can be partitioned into alternating segments corresponding to cuts through unoccupied and occupied areas of the BL packing. We shall show that the sum of the occupied segments is at least the sum of the unoccupied segments. For convenience we may restrict ourselves to lines which do not coincide with the (upper or lower) edges of any piece. Since the set of such lines is of measure zero, ignoring them will not influence our claim that A is at least half occupied.

Initially, consider the partition of a given line just prior to when the first rectangle, say q, is assigned with a lower edge at a height exceeding the height, h, of the line. The piece q need not be in A; its existence is guaranteed by the fact that there is a piece packed above A. We claim that at that point in the assignment sequence the line is "half occupied".

First, bottom-up packing implies that all lines must cut through at least one piece. Second, all lines must cut through a piece abutting the left bin edge. For suppose not; then the left-most piece, say q', cut by the line must abut another piece, say q", to the left and entirely below the line. Thus, the length, x, of the unoccupied, initial segment of the line must be at least the width of q". But since q" was packed prior to q, the width of q must be less than that of q" and hence less than x. Since at the point in time we are considering, no piece has been assigned entirely above the line, the space vertically above the initial segment must be completely unoccupied. Thus, we have the contradiction that q would have fit into the space above q" in such a way that its lower edge is at a height less than h.

Now consider any segment, S, of the line which cuts through an unoccupied space. Let p be the piece bordering S on the left. Since when q is assigned, it is placed above the line, q must be wider than the length of S. (Once again, at the time q is assigned its height could not prevent its placement in a sufficiently wide unoccupied space cut by the line.) But q is packed later than p; consequently, p is at least as wide as q. It follows that for each segment representing unoccupied space along the line there is a longer segment representing occupied space immediately to its left. Clearly, for any given line, the sum of the segment lengths corresponding to unoccupied space must be monotonically non-increasing as the packing sequence progresses. Therefore, the line continues to be at least half occupied. Finally, "integration" over the height of A verifies that A is at least half full. □

Corollary 1

If L in the statement of Theorem 3 consists only of squares then

$$\frac{h_{BL}}{h_{OPT}} \leq 2$$

This bound is best possible in the sense of Theorem 2.

Proof: First, define A' as the area extending from height y to height $h_{BL} - y$, where y is the size of the tallest rectangle (now square) assigned above A in Theorem 2. Let p denote this square. As in Theorem 2, $A' \subset A$ is shown to be at least half occupied. But now, if w denotes the width of the bin, we observe that the cumulative occupied area of the upper and lower $w \times y$ slabs of the packing is at least wy and hence they are (when considered together) half occupied. This follows from the facts that p is at most as large as any square on the bottom of the bin, the bottom of the bin is full except possibly for a space at the right end, and the area of p must exceed y times the width of this space. Hence, the entire packing is at least half occupied from 0 to h_{BL}. If follows immediately that $h_{BL} \leq 2h_{OPT}$. □

We have seen that a BL algorithm can yield reasonably good packings when the list of pieces is sorted into decreasing order by width. That is, Theorem 3 shows that the bin height used is no more than three times the optimal bin height, and if the pieces are squares, then the packing can be no more than twice as high.

A natural question to ask next is, "For every set of pieces, is there some ordering of those pieces into a list such that the BL rule, when applied to that list, yields an optimal packing?" Our checkerboard example, using squares, which showed that a list sorted into decreasing order by size can use up to twice as much space as an optimal packing, can be packed optimally by the BL algorithm if the list is sorted into increasing order by size. While it might be difficult in practice to actually determine an ordering for which the BL rule produces an optimum packing, it would be comforting to know that one was not excluding the possibility of finding an optimum packing by considering only bottom-up packings.

Unfortunately, there are sets of pieces for which no BL packing is optimal. That is, no matter what ordering is used, the BL algorithm will produce a suboptimal packing. In fact we shall present an example using squares only which demonstrates that an optimal packing can be as little as 11/12 the height of the best bottom-up packing.

Theorem 4. There exist sets of squares such that the ratio of the bin height used by the best bottom-up packing to that of an optimum packing is at least $12/(11+2\varepsilon)$ for any sufficiently small $\varepsilon > 0$.

Proof: Consider the set of squares of sizes (6,6,5,5,4,4,3,1,1) and a rectangle of width 15. An optimum packing, of height 11, is shown in Fig.

4. We first demonstrate that (up to obvious left-right symmetries) this is the only optimum packing, and then we modify the example slightly to obtain the theorem.

Since Fig. 4 is a tight packing, any optimum packing must have height 11. For an arbitrary optimum packing consider the 15×11 rectangle that it packs to be divided into 15 disjoint 1×11 vertical slabs. Let the type of a slab be an ordered list of the sizes of the squares that the slab intersects. The only possible types are:

(a) 6-5
(b) 6-4-1
(c) 6-3-1-1
(d) 5-5-1
(e) 5-4-1-1
(f) 4-4-3

Let a denote the number of slabs of type (a), etc. The following equations must then hold, for a,b,c,d,e,f nonnegative integers:

$$\begin{array}{lcl} a+b+c+d+e+f & = & 15 \\ a+b+c & = & 12 \\ a \quad\quad +2d+e & = & 10 \\ \quad b+ \quad\quad e+2f & = & 8 \\ \quad\quad c \quad\quad +f & = & 3 \\ \quad b+2c+d+2e & = & 2 \end{array}$$

The first equation reflects the fact that there is a total of 15 slabs; the remaining five equations account for the presence of the squares of sizes 6,5,4,3,1 respectively. For example, the second equation states that the presence of exactly two 6×6 squares requires 12 slabs of types with 6's in them. By the last equation we have that either $c = 0$ or $c = 1$. Choosing $c = 1$ yields a contradiction, and $c = 0$ gives us the unique solution to the above equations:

$a = 10$
$b = 2$
$c = d = e = 0$
$f = 3$

Any solution having the slab types in the above numbers must either look like Fig. 4 or its reflection. This can be proved by observing that (since $a = 10$), each 5×5 is entirely above or below a 6×6. Take a particular 5×5 and consider the slab which intersects the adjacent 6×6 but does not contain the 5×5. This must be of type 6-4-1 or 6-5 (using the other five). The 6-5 possibility can't happen (since there would be a 6×3 unfilled space next to the other 6×6), and the pieces of the 6-4-1 must occur in the order 6,1,4 to prevent an unfillable gap between the 4×4 and the edge. The 4-4-3 slabs must come next: the pieces must be in the order 4,3,4 to leave room for the 6×6. Finally, the 6-4-1 and 6-5 slabs complete the picture. Thus, Fig. 4 represents the only way to pack the given squares into a 15×11 rectangle.

We now modify things so that (i) the 3×3 now has size $(3+\varepsilon) \times (3+\varepsilon)$ and (ii) the bin now has width $15+\varepsilon$. The preceding proof shows that (to within ε) the packing in Fig. (i) is still optimum. However, we note that in the modified packing there must be gaps on the bottom row between the 5×5 and 4×4 and between the 4×4 and 6×6 which add up to ε; otherwise the 1×1 and $(3+\varepsilon) \times (3+\varepsilon)$ will not be able to fit on top of the 4×4. Since no BL rule can produce those gaps, the optimum packing of Fig. 4 is unachievable. The best that such an algorithm can do is a packing of height 12. Thus we have

$$\frac{h_{BL}}{h_{OPT}} \geq \frac{12}{11+\varepsilon}$$

for this example. □

III. CONCLUSIONS

This paper is but a beginning in the study of fast, effective approximation algorithms for packing pieces in two dimensions. We have seen that, although performance of these algorithms can be very poor, simple measures such as ordering on piece size can produce algorithms with much more reasonable performance. Indeed, with such algorithms it appears that worst-case performance can only be approached by essentially pathological cases.

Many open problems related to our models remain for future research. Questions that should be resolved are those connected with the specialization to squares and those arising from list orderings we have not considered. For example, what is a tight bound for the special case of increasing squares? Another question concerns the implementation of the BL algorithms. The complexity of such algorithms for decreasing widths is open. How does one efficiently maintain the structure of available space as the packing sequence progresses?

REFERENCES

[1] Codd, E. G., "Multiprogram Scheduling," Comm. of the ACM, Parts 1 and 2, (1960), pp. 347-350, Parts 3 and 4, 3 (1960), pp. 413-418.

[2] Garey, M. R. and R. L. Graham, "Bounds on Multiprocessing Scheduling with Resource Constraints," SIAM J. on Comp., 4 (1975), pp. 187-200. (See also Chapter 4 in reference 7).

[3] Erdős, P. and R. L. Graham, "On Packing Squares with Equal Squares," Tech. Rep. STAN-CS-75-483, Computer Science Dept., Stanford University, March 1975.

[4] Kleitman, D. J. and M. M. Krieger, "An Optimal Bound for Two-Dimensional Bin-Packing," Proc. 16th Annual Symposium on Foundations of Computer Science, Oct. 1975.

[5] Kleitman, D. and M. Krieger, "Packing Squares in Rectangles, I," Annals of the New York Academy of Sciences, Vol. 175, Article 1, pp. 253-262, July 1970.

[6] Meir, A. and L. Moser, "On the Packing of Squares and Cubes," Journal of Combinatorial Theory, 5 (1968), pp. 126-134.

[7] Ullman, J. D., "Complexity of Sequencing Problems," Computer and Job-Shop Scheduling Theory, E. G. Coffman, Jr. (ed), John Wiley and Sons, 1975.

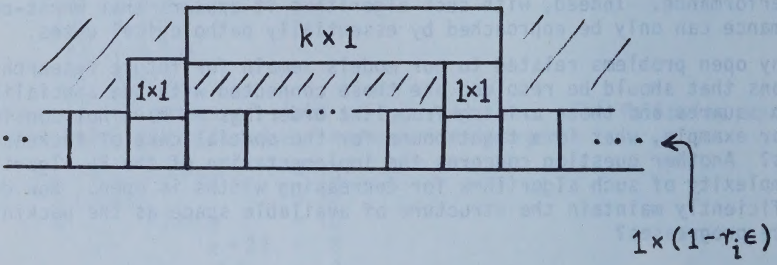
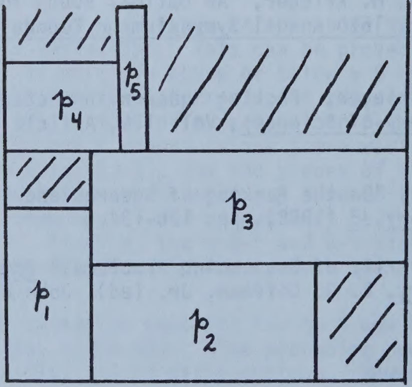

k × 1
1×1
1×1
...
...
...
1 × (1-r_iε)

Figure 2 - The Increasing Widths Example


p_5
p_4
p_3
p_1
p_2

$L = (p_1, p_2, p_3, p_4, p_5)$

x_i: 3 8 11 4 1

y_i: 6 4 4 3 5

Figure 1 - Two-Dimensional Packing

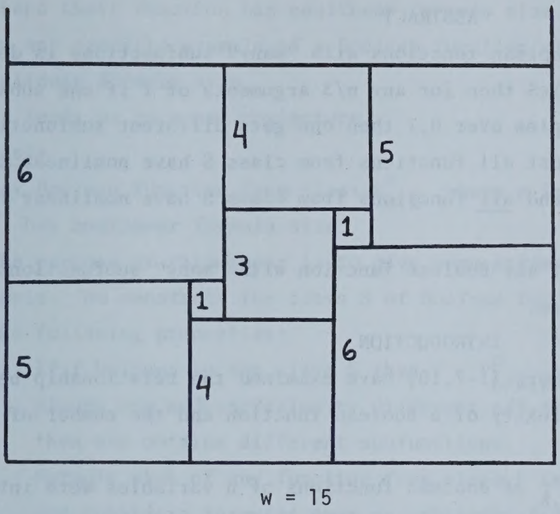

4
5
6
1
3
1
5
4
6
w = 15

Figure 4 - Optimum vs. Best BL Packings

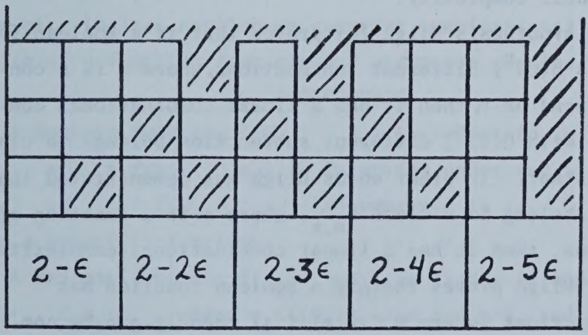

2-ε
2-2ε
2-3ε
2-4ε
2-5ε

Figure 3 - The Checkerboard Example

A FAMILY OF BOOLEAN FUNCTIONS WITH NONLINEAR FORMULA SIZE

YURI BREITBART*
Electrical Engineering & Computer Science Department
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin 53217

ABSTRACT

A class S of Boolean functions with "many" subfunctions is defined. In fact if $f(x_1,\ldots,x_n)\varepsilon S$ then for any n/3 arguments of f if one substitutes different n/3-tuples over 0,1 then one gets different subfunctions.

We prove that almost all functions from class S have nonlinear combinational complexity and all functions from class S have nonlinear formula size.

We conjecture that any Boolean function with "many" subfunctions has a nonlinear formula size.

INTRODUCTION

Recently many authors [1-7,10] have examined the relationship between the combinational complexity of a Boolean function and the number of its distinct subfunctions.

In [1] classes $P^n_{m,k}$ of Boolean functions of n variables were introduced such that if we replace any m variables $(0 \leq m \leq n)$ by every m-tuples over (0,1) then we obtain at least k different subfunctions. The authors examined the class $P^n_{3,5}$ and demonstrated that the minimal combinational complexity for the functions from $P^n_{3,5}$ must be no less than $(7/6)n-2/3$. There was a hypothesis that if a Boolean function belongs to a class $P^n_{m,k}$ (where k grows as $0(2^m)$ and m is a function of n, then this function has a non-linear combinational complexity.

In support of the hypothesis Uligh [6] proved that if a Boolean function of n variables has $0(c2^n)$ different subfunctions, where c is a constant that does not depend on n, then it has a linear combinational complexity. Any function with $0(c2^n)$ different subfunction belongs to class $P^n_{m,a}$, where a is a constant. In other words Uligh has demonstrated that if a function does not belong to a class $P^n_{m,k}$, where m is a function of n, and k grows when n grows, then it has a linear combinational complexity.

In next paper [7] Uligh proves that if a Boolean function has $\gamma(n)3^n$ different subfunctions (where $0 < \gamma(n) < 1$) then it can be computed with combinational complexity $A(\gamma)\cdot 2^n/n$, where $0 < A(\gamma) < 1$. This is essentially smaller than $2^n/n$ the combinational complexity of the most complex Boolean function.

---

*This work was supported in part by NSF Grant GJ-42907 and by NSF Grant MCS77-06321.

This was another strong evidence in support of the hypothesis.

However the above mentioned hypothesis was finally contradicted by A. Meyer and M. Patterson. They have exibited a Boolean function from class $P^n_{m,2^m}$, where m=n/logn that has a linear combinational complexity. On the other hand their function has nonlinear formula size. Moreover as far as we know any specific example of a Boolean function with many subfunctions has nonlinear formula size.

It leads us to a new conjecture.

Hypothesis

Any Boolean function from class $P^n_{m,k}$, where m is a function of n and $k=0(2^m)$ has nonlinear formula size.

The purpose of this paper is to give some arguments in support of our hypothesis. We construct the class S of Boolean functions of n variables with the following properties:

a. If f belongs to the class S then $f \varepsilon P^n_{n/3,2^{n/3}}$ i.e. if one replaces any n/3 variables by different n/3-tuples of 0,1 then one obtains different subfunctions.

b. Formula size of any function from class S is nonlinear (in n) if one considers formulas over an arbitrary finite basis.

c. Formula size of any function from class S grows at least as $0(n^2)$ if one considers formulas over v, &, $^-$.

d. Almost all functions from class S have nonlinear combinational complexity. In fact if $c(n)=\max_{f\varepsilon S} c(f)$ then $c(n) \sim n^2/6\log n$.

In next section we give definitions and formulate some facts that will be used in proving a) - d). In third section we define the class S and prove a). In last section we consider combinational complexity and formula size of a function from class S and prove results b) - d).

BACKGROUNDS

A Boolean function $f(x_{1},\dots,x_{n})$ is symmetric in $(x_{i_1},\dots,x_{i_k}) \subseteq (x_{1},\dots,x_{n})$ if $f(x_{1},\dots,x_{n})$ is invariant under any permutation of $x_{i_1},\dots,x_{i_k}$. If $f(x_{1},\dots,x_{n})$ is symmetric in $(x_{1},\dots,x_{n})$, then it is said to be a symmetric function.

A Boolean function $f(x_{1},\dots,x_{n})$ is linear if and only if $f(x_{1},\dots,x_{n})= c_1 x_1+\dots c_n x_n + c_{n+1}$ where $c_i \varepsilon ((0,1)$, $1 \leqslant i \leqslant n+1$, and + is sum of modulo two.

We will use the following well-known fact.

Proposition 0

If $f(x_{1},\dots,x_{n})$ is symmetric and linear then either

$f(x_{1},\dots,x_{n}) = x_1+\dots+x_n+c$

or $f(x_{1},\dots,x_{n}) = c$, where $c\varepsilon(0,1)$.

If α is a tuple over (0,1), $|\alpha|$ denotes the number of one's in α and $\ell(\alpha)$ the number of components in α. If A is a set, then $||A||$ denotes the number of elements in A.

Let $S_n^i(x)$ be a symmetric Boolean function such that $S_n^i(x)=1$ if and only if $|x| = i$, where x is an n-tuple over (0,1).

Proposition 1

If $f^1(x,y) = \sum_{i=0}^{n} L_i^1(x)\cdot S_n^i(y)$ and $f^2(x,y) = \sum_{i=0}^{n} L_i^2(x)\cdot S_n^i(y)$,1) where $L_i^j(x)$ $(1 \leqslant j \leqslant 2,\ 0 \leqslant i \leqslant n)$ is a linear function, $x=(x_1,\dots,x_n)$, $y=(y_1,\dots,y_n)$, and the x's and y's are disjoint variables. Then $f^1 = f^2$ if and only if for all i, $0 \leqslant i \leqslant n$, $L_i^1 = L_i^2$.

Proof easy.

Proposition 2

If $f = \sum_{i=0}^{n} L_i(x)\cdot S_n^i(y)$, where $L_i(x)$ $(0 \leqslant i \leqslant n)$ is a linear function of x and the x's and y's are distinct. Then, if x_k is an essential variable of at least one L_i, then x_k is an essential variable of f.

Proof easy.

CLASS S

Now we define a class S of Boolean functions of n variables such that $S \subset P^n_{n/3,2^{n/3}}$.

We consider three pairwise-disjoint sets of variables: $x^1 = (x_{11},\dots,x_{1k})$; $x^2 = (x_{21},\dots,x_{2k})$, $x^3 = (x_{31},\dots x_{3k})$, where n is even and more than one.

Let $(L_i^j)_{i=1}^{k}$, $(1 \leqslant j \leqslant 3)$ be three systems of linear independent (in GF(2)) linear functions of k variables.

Let

$$f^1(x^1,x^2,x^3) = \sum_{i=1}^{k} L_i^1(x^1)\cdot\left(S_{2k}^{i-1}(x^2,x^3) + S_{2k}^{k+i}(x^2,x^3)\right) \tag{1}$$

$$f^2(x^1,x^2,x^3) = \sum_{i=1}^{k} L_i^2(x^2)\cdot\left(S_{2k}^{i-1}(x^1,x^3) + S_{2k}^{k+i}(x^1,x^3)\right) \tag{2}$$

$$f^3(x^1,x^2,x^3) = \sum_{i=1}^{k} L_i^3(x^3)\cdot\left(S_{2k}^{i-1}(x^1,x^2) + S_{2k}^{k+i}(x^1,x^2)\right) \tag{3}$$

$$F(x^1,x^2,x^3) = f^1+f^2+f^3 \tag{4}$$

Function F will belong to the class S. Thus for any three systems of linear independent linear functions we construct exactly one function that belongs to S. Therefore the number of different functions in S is equal to N_k-number of triples of k linear independent linear functions each of which depends on k variables. On the other hand one can see that

---

1) Σ represents addition modulo two.

$$N_k \geqslant \left((2^k-1)(2^k-y)\ \ldots\ (2^k-2^{k-1})\right)^3 \geqslant 2^{3k^2}$$

Thus we prove

Lemma 3

$||S|| \geqslant 2^{3k^2}$.

To prove the result mentioned in the beginning of this paragraph we will need few lemmas. First let us pick any n variables of F: $x_{1i_1} \ldots x_{1i_{k1}}$, $x_{2j_1}, \ldots x_{2j_{k2}}, x_{3r_j}, \ldots x_{3r_{k3}}$, where $k1+k2+k3 = k$.

We define $\tilde{x}^p = x^p \setminus (x_{pi_1}, \ldots x_{pi_{k1}})$, where $1 \leqslant p \leqslant 3$.

It is clear that $||\tilde{x}^1|| = k2+k3$, $||\tilde{x}^2|| = k1+k3$ and $||\tilde{x}^3|| = k1+k2$.

Let $\alpha = \alpha^1\alpha^2\alpha^3$ and $\beta = \beta^1\beta^2\beta^3$, where $\ell(\alpha) = \ell(\beta) = k$ and $\alpha^1, \alpha^2, \alpha^3$ $(\beta^1, \beta^2, \beta^3)$ are k1-, k2-, k3-tuples of 0's, 1's respectively.

Let: $t_{11} = |\alpha^1| + |\alpha^2|$, $t_{12} = |\alpha^1| + |\alpha^3|$, $t_{13} = |\alpha^2| + |\alpha^3|$,

$t_{21} = |\beta^1| + |\beta^2|$, $t_{22} = |\beta^1| + |\beta^3|$, $t_{23} = |\beta^2| + |\beta^3|$

Lemma 4

$t_{1i} \neq t_{2i}$ if and only if $f^{4-i,\alpha} + f^{4-i,\beta}$[2] depends essentially on at least one of the variables in $\tilde{x}^{4-i}$.

Proof

Without loss of generality assume i=1 and $t_{11} \leqslant t_{21}$. We consider the system of linear independent linear functions $(L_i^3(x))_{i=1}^k$ and decompose each of the L_i^3 as follows:

$$L_i^3(x^3) = \tilde{L}_i(\tilde{x}^3) + L_i^*(x^*) \quad (x^* = x^3 \setminus \tilde{x}^3)$$

Then one can easily prove that

$$f^{3,\alpha} + f^{3,\beta} = \sum_{i=t_{11}+1}^{k+k3+t_{11}+1} (\tilde{L}_i(\tilde{x}) + \tilde{L}_{i+t_{21}-t_{11}}(\tilde{x}) + L_i^*(\alpha^3) + L_{i+t_{21}-t_{11}}^*(\beta^3)) \cdot$$

$$S_{k+k3}^{i-t_{11}-1} \cdot (\tilde{x}^1, \tilde{x}^2) \tag{5}$$

where indices for the L's are considered modulo (k+1) and $L_{k+1} = 0$ by definition.

Now, if $t_{11} = t_{21}$, then $f^{3,\alpha} + f^{3,\beta}$ does not depend essentially on any variable of x^3, as can be seen from (5). If $t_{11} < t_{21}$, we will show that there exists at least one i such that $\tilde{L}_i^3 + \tilde{L}_{i+t_{21}-t_{11}}^3 \neq 0$.

If $\tilde{L}_i^3 + \tilde{L}_{i+t_{21}-t_{11}}^3 = 0$ for all i, then $\tilde{L}_p^3 = \tilde{L}_m^3$ whenever $p \equiv m \pmod{t_{21}-t_{11}}$. Since $L_{k+1}^3 = 0$, $(\tilde{L}_i^3)_{i=1}^k$ must have rank $r < t_{21}-t_{11}$, but since $(L_i^3)_{i=1}^k$ is a linear independent system, it has rank $||\tilde{x}^3||$. Since $t_{21}-t_{11} \leqslant ||\tilde{x}^3||$, this is a contradiction. Therefore, there are at least one i such that $\tilde{L}_i^3 + \tilde{L}_{i+t_{21}-t_{11}}^3 \neq 0$.

---

[2] By $f^{i,\alpha}$ (i=1, 2, 3) and F^α respectively we will denote the result of substituting α instead of chosen k variables, where $\alpha = \alpha^1\alpha^2\alpha^3$.

By proposition 2, it follows that $f^{3,\alpha} + f^{3,\beta}$ depends essentially on the essential variables of $\tilde{L}_i^3 + \tilde{L}^3_{i+t_{21}-t_{11}}$. Q.E.D.

Lemma 5

If $F^\alpha = F^\beta$, then $f^{i,\alpha} + f^{i,\beta}$ is symmetric in $\tilde{x}^i$.

Lemma 6

If $f^{i,\alpha} + f^{i,\beta}$ is symmetric in $\tilde{x}^i$ then

$$f^{i,\alpha} + f^{i,\beta} = \left(\sum_{x_j \varepsilon \tilde{x}^i} x_j\right) \cdot R^{i1}(\tilde{x}^{i+1}, \tilde{x}^{i+2}) + R^{i2}(\tilde{x}^{i+1}, \tilde{x}^{i+2}) \qquad (6)$$

where $1 \leq i \leq 3$ and R^{i1}, R^{i2} are symmetric functions and + is sum of modulo three.

Lemma 7

If $\sum_{i=1}^{3} (f^{i,\alpha} + f^{i,\beta}) = 0$, then each R^{i1} $(i=1,2,3)$ must be a constant.

Lemma 8

If $f^{4-i,\alpha} + f^{4-i,\beta}$ is symmetric in $\tilde{x}^i$ then R^{i1} equals zero if $t_{1i}=t_{2i}$.

Proof

Follows from lemma's 4 and 6.

Lemma 9

For any α^i and β^i, $f^{i,\alpha} + f^{i,\beta} \neq 1$

Theorem 10

If $\alpha \neq \beta$, then $F^\alpha \neq F^\beta$.

Proof

Let $\alpha \neq \beta$, but assume that $F^\alpha = F^\beta$. By lemma 5 $f^{i,\alpha} + f^{i,\beta}$ $(i=1,2,3)$ is symmetric in $\tilde{x}^i$. Therefore, by lemma 6 we conclude that for $f^{i,\alpha} + f^{i,\beta}$ (6) holds. Also, $\sum_{i=1}^{3} (f^{i,\alpha} + f^{i,\beta}) = 0$. By lemma 7 we conclude that each R^{i1} is a constant function. Using arguments similar to lemma 7 one can conclude that R^{i2} must be a linear function. For the same reasons no R^{i2} can be linear symmetric and non-constant. Therefore R^{i1} is equal to zero, and so by lemma 8, $t_{11} = t_{21}$; $t_{12} = t_{22}$ and $t_{13} = t_{23}$.

But if each $R^{i1} = 0$, then by lemma 9, we conclude that R^{i2} cannot be the constant one. Therefore $R^{i2} = 0$ $(i = 1,2,3)$. Hence $f^{1,\alpha} = f^{1,\beta}$. By the linear independence of $(L_i^j)_{i=1}^k$, $(1 \leq j \leq 3)$ we can conclude that $\alpha_x = \beta_x$ $\alpha_y = \beta_y$ and $\alpha_z = \beta_z$. Contradiction.

FORMULA SIZE AND COMBINATIONAL COMPLEXITY

Here we consider complexity results for functions of class S. We begin with presenting of three results about formula size of functions from S.

First we prove a result about formula size of functions from S over an arbitrary finite basis B.

Theorem 11

Let B be a Boolean finite basis and $F \in S$.

Then formula size of F is nonlinear (in N-number of variables of F).

Proof

We will use Specker's theorem [8] that will be formulated here for a reader's convenience.

Let $f(x_{1,\dots} x_n)$ be a Boolean function of n-arguments and $L_B(f)$ the formula size of f over a finite basis B.

Specker's Theorem [12]

If $L_B(f) < cn$ for some constant c, then for any integer m, if $n > \phi_B(m,c)$, a subset $x^* = \{x_{i_1}, \dots, x_{i_m}\}$ of the arguments of f can be found such that

$$f^{\tilde{x}}_{\tilde{0}} = c_o + c_1 \prod_{p=1}^{m} x_{j_k} + c_2 \sum_{p=1}^{m} x_{j_p} \quad \text{where } \tilde{x} = x \setminus x^* \tag{7}$$

Let F be a function from S. Then

$$F = \sum_{i=1}^{k} L_i^1(x^1)\left(S_{2k}^{i-1}(x^2, x^3) + S_{2k}^{k+i}(x^2, x^3)\right) + \sum_{i=1}^{k} L_i^2(x^2)\left(S_{2n}^{i-1}(x^1, x^3) + S_{2k}^{n+i}(x^1, x^3)\right) + \sum_{i=1}^{k} L_i^3(x^3)\left(S_{2k}^{i-1}(x^1, x^2) + S_{2k}^{k+i}(x^1, x^2)\right)$$

Since $\{L_i^1(x^1)\}$ is a linear independent system of linear functions over GF(2) such $\alpha_1 \cdots \alpha_k$ can be found that

$$H = F^{\tilde{\alpha}}_{x^1} = \overline{S}^{k}_{2k}(x^2, x^3) + \sum_{j=o}^{k} L_{i_j}^2(x^2) \cdot S_k^j(x^3) + \sum_{j=o}^{k} L_{p_j}(x^3) \cdot S_k^j(x^2) \tag{8}$$

Since $\left(L_j^2(x^2)\right)_{j=1}^{k}$ is linear-independent system of linear forms, one can find $\beta^2 = \beta_1 \dots \beta_k$ such that $L_1^2(x^2)^{\beta^2}_{x^2} = 1$, $L_2^2(x^2)^{\beta^2}_{x^2} = 1$ and for any other

$j \neq 1,2$ $L_j^2(x^2)^{\beta^2}_{x^2} = 0$

Let r be a number of ones in k-tuple β^2. Then $G(x^3) = H^{\beta^2}_{x^2} =$ $S^{k-r}_{\overline{k}}(x^3) + S_k^i(x^3) + S_k^1(x^3) + L_{p_r}(x^3)$

Now we can apply Specker's theorem. Let m=5 and $n > \phi_B(5,c)$. Let $\tilde{w} = \{x,y,z,u,v\}$ be a set of five arguments of G.

Then

$$R(x,y,z,u,v) = G(x^3)^{x^3 \setminus \tilde{w}}_{\tilde{0}} = S_5^1 + S_5^2 + S_5^{5-r} + L_{p_r}(x,y,z,u,v)$$

It is not difficult to see that R(x,y,z,u,v) under no conditions can be expressed in the form (7). Therefore $G(x^3)$ has nonlinear (in k) formula size over the basis B.

Next theorem gives a better lower bound on formula size of functions from S.

Theorem 12 Let $B^1 = (v, \&, +, ^{-})$

If $F \varepsilon S$ then $L_{B^1}(F) \geqslant 0\,(n\log n/\log\log n)$.

Proof

As is in theorem 11 we will find such α^1 that $\phi = F^{\tilde{\alpha}}_{x^1}$ has a form (8). Now we will pick values for $y_1 \ldots y_n$ in such a way that $\Sigma L_{2i_j}(y)\, S_n^j(z)\Big|^{B}_{\tilde{y}} = C_r^n$ for some constant r. Again, since $\{L_{2_i}\}$ is linear independent system we can find such values for x^2. As a result of these two steps we obtain a function

$$G(x^3) = S_k^{r1}(x^3) + C_r^n(z) + L_{3i_p}(z) + 1$$

Note that any linear function derived from $G(x^3)$ by substituting constants is a function of less than r-variables, (i.e. one has to substitute by constants at least (k-r) variables).

Applying Fisher-Meyer-Patterson result [9] we derive

$$r > k/k(a\log k)^{2m-1} \qquad m > \log k/c\log\log k$$

Hence $L(G) \geqslant k\log k/c\log\log k$ and finally $L(F) \geqslant L(G) > n\log n/c\log\log n$. It is interesting to compare this lower bound with upper bound for functions from S. Khrapchenko [13] proves that formula size of an arbitrary elementary symmetric function of n variables is no more $n^{1+\varepsilon_p}$ ($\varepsilon_p \to 0$, when $p \to \infty$) for some finite Boolean basis B. Therefore formula size of a function from S is no more than $n^{2+\delta_p}$ ($\delta_p \to 0$ when $p \to \infty$) over B and at least nonlinear as follows from theorem 11.

For Basis B we can prove

$$0\,(n\log n/\log\log n) \leqslant L_{B^1}(F) \leqslant 0(n^{3.56})$$

where upper bound follows from unpublished Pippenger's result.

Next theorem can be proven by using arguments similar to that were used in theorems 11, 12.

Theorem 13

If $B^2 = (v, \&, ^{-})$ then $L_{B^2}(F) \geqslant 0(n^2)$ for any function $F \varepsilon S$.

We consider two results concerned with combinational complexity of a function $F \varepsilon S$.

Let c(F) be a combinational complexity of a function f.

Let $c(n) = \max\limits_{F \varepsilon S} c(f)$

Theorem 14

$c(n) \geqslant n^2/6 \log n$ and moreover for "almost all" functions $F \varepsilon S$ [3)]

$$c(f) \geqslant n^2/6 \log n$$

Proof follows using counting arguments and lemma 3.

Although almost all functions in S have nonlinear combinational complexity, it is easy to construct function ϕ, such that $c(\phi) \leqslant rn$, where r is a constant independent of n.

It is also worth mentioning that $\lim_{n\to\infty} c(n)/(n^2/6\log n) = 1$

Using the Nechiporuk [12] construction to compute a system of linear forms we prove

Theorem 15

$$c(n) \leqslant n^2/6\log n.$$

CONCLUSIONS

In this paper interdependancy between number of subfunction of a Boolean function and its formula size and combinational complexity was considered. We claimed that if a Boolean function has "many" subfunctions that it has nonlinear formula size. To support this claim we constructed a class of Boolean function consisting of more than $2^{n^2/3}$ different functions that satisfies to following conditions.

a) If $F \varepsilon S$ then replacing any n/3 arguments of F by different n/3-tuples of 0's and 1's one obtains different subfunctions.

b) Any F from S has nonlinear formula size over arbitrary finite basis and over basis of $f, \&, +^-$ consisting of two arguments and formula size grows at least as $0\ (n\log n/\log\log n)$.

c) Any F from S has formula size over basis $v, \&, ^-$ that grows at least as $0(n^2)$.

d) Almost all functions from S have nonlinear combinational complexity. In fact $c(n) \sim n^2/6\log n$, where $c(n) = \max_{F \varepsilon S} c(F)$.

REFERENCES

1. L. H. Harper, W. Hsie, T. E. Savage. "A class of Boolean function with linear combinational complexity," Theoretical Computer Science, 1, 2, pp. 161-183.

[3)] We will say that a property P is true for "almost all" $F \varepsilon S$ if $\lim_{n\to\infty} N^n_p/N_n = 1$, where N^n_p is the number of functions in S, having the property P, and N_n is the number of functions in S_n.

2. L. H. Harper. "An nlogn lower bound on synchronous combinational complexity." Transaction Amer. Math. Society, 1976.

3. L. H. Harper, T. E. Savage. "Nonlinear lower bounds on the complexity of integer and matrix multiplication." To appear.

4. A. L. Toom. "The complexity of combinational circuits to compute Boolean functions with small number of subfunctions." Syst. Theory Research, vol. 18 (1968) pp. 77-84.

5. P. Denew. "On one function that is hard to compute by Cascade method." Mathematical Notes, 16,5 (1975).

6. Uligh. "On a connection between combinational complexity and number of subfunctions of a Boolean function." Problemy Kibernetiki, 26, pp. 183-202, 1973. (Russian).

7. Uligh. "On one family of classes of Boolean functions that have simple combinational complexity." Problemy Kibernetiki, 28, pp. 25-42, 1974. (Russian).

8. L. Hodes, E. Specker. "Length of formulas and elimination of quantifiers I." Contributions to Mathematical Logic, North Holland Publ. Co., 1968, 175-188.

9. M. T. Fisher, A. R. Meyer, M. S. Patterson. "Lower bounds on the size of Boolean formulas." Proceedings of Seventh Annual ACM Symposium on Theory of Computing, 1975, 137-44.

10. Y. J. Breitbart. "Comparison of complexities of realizations Boolean functions by automata and turing machines," Soviet Physics Dokl. 13, 1968.

11. Nechiporuk. On gate schemas," Doklady Akademii Nank SSSR, 148, 1, 1963.

12. Vilfan. "The complexity of finite functions." Project MAC, Cambridge, Mass., 1972.

13. Khrapchenko. On complexity of symmetric function Problemy Kybernefiki, 31, 1974.

A SOLUTION FOR BILINEAR DISCRETE-TIME CONTROL SYSTEMS

K. N. SWAMY and T. J. TARN
Systems Science and Mathematics
Washington University
St. Louis, Missouri 63130

ABSTRACT

Optimal control of a class of time invariant single-input, discrete bilinear systems is investigated in this paper. Both deterministic and stochastic problems are considered for control unbounded and the cost functional quadratic in state.

In the deterministic problem, for x_o in a certain Set Σ_o optimal path has been extracted via the functional equation of dynamic programming from a subcollection of paths, and it is shown to be the optimal in the totality of paths from x_o. The solution then is the same as the solution to the associated linear system. The optimal path may be a regular path or a singular path.

Stochastic control problem is considered with perfect state observation, and additive and multiplicative noise in the state equation. It is demonstrated that the presence of noise simplifies the analysis compared to that in the deterministic case by virtue of integration over certain sets of measure zero. In particular, it is seen qualitatively that even in the case when optimum does not exist for the deterministic problem, it exists for the corresponding stochastic control problem.

I. INTRODUCTION

The purpose of this paper is to investigate the question of optimal control of a class of time-invariant, single input discrete bilinear systems. In [1] are given examples of a large number of problems which can be modelled as bilinear systems. Several aspects like controllability [2,3,4], structural properties and realization [6-10] of bilinear systems have been explored in the literature. A state-of-the-art survey is given in [5].

For the deterministic optimization problem, solutions have been obtained for the free end point problem. A procedure has been developed to construct the initial set Σ_o and the collection of paths $P'(x_o)$, x_o in Σ_o, which is such that the optimal in $P'(x_o)$ is also the optimal in the totality of paths from x_o. The optimal path has been extracted using the functional equation of dynamic programming. The optimal path then is the same as the solution to the linear system, associated with the bilinear system, and it is a regular path or a singular path.

The special case in which the regular path exists is examined. Then Σ_o is the set of states for which the regular path exists.

The optimal return function is seen to be discontinuous, in general, across the hyperplane H_i for each i and the nonexistence of optimum for $x_o \notin \Sigma'_o$ is seen to be essentially due to this discontinuity.

In the stochastic optimization problem, complete state information is assumed available. Both additive and multiplicative noise are considered. Solution to the optimal control problem has been obtained using the method of dynamic programming. The principal contribution of this section consists in showing that the presence of noise simplifies the analysis compared to that in the deterministic case by virtue of integration over certain sets of measure zero. The effects of discontinuities of the optimal cost function which are confined to sets of measure zero are annihilated by the noise in the stochastic case, while they propagate in the deterministic case.

II. DETERMINISTIC OPTIMAL CONTROL

2.1. Problem Statement

Consider the discrete-time bilinear system described by

$$x_{k+1} = Ax_k + u_k Bx_k, \quad 0 \leq k \leq N-1, \quad x_o = x(t_o) \tag{2.1}$$

where $x_k \in R^n$ is the state vector, $u_k \in R$ is the scalar control. The $n \times n$ constant matrices A and B are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & . & . & . & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & . & . & . & 1 \\ a_1 & a_2 & . & . & . & . & a_n \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & . & . & . & 0 \\ 0 & 0 & . & . & . & 0 \\ \vdots & & & & & \\ b_1 & b_2 & . & . & . & b_n \end{bmatrix} \tag{2.2}$$

The problem is to choose the sequence of controls $u_o, \ldots, u_{N-1}$ such that the cost functional

$$J = \sum_{k=1}^{N-1} x_k^T Q x_k + x_N^T Q' x_N$$

is a minimum, subject to (2.1). Q and Q' the $n \times n$ constant cost matrices are symmetric and nonnegative. N is a fixed positive integer. T denotes the transpose.

Remark 2.1. It is shown that [2] if rank B = 1 and if (A,c) is a controllable pair, where $B = cb^T$, then (2.1) can be transformed to the canonical form with A,B as specified in (2.2) by coordinate transformations.

The following definitions are made in connection with the bilinear system.

Definition 2.1. The optimal path is called a regular path if the optimal control is defined uniquely at each k.

Definition 2.2. A singularity is said to exist at k if the optimal path is such that the optimal control is nonunique at k.

Definition 2.3. Optimal path is called a singular path if it has at least one singularity.

2.2. Preliminary Results

Define for any symmetric $n \times n$ matrix

$$Q_k = \left(q_{ij}^k\right) \text{ with } q_{nn}^k \neq 0, \text{ a symmetric matrix}$$

$$C_k = \left(c_{ij}^k\right)$$

$$c_{ij}^k \triangleq \frac{\left(q_{nn}^k a_i + q_{i-1,n}^k\right)\left(q_{nn}^k a_j + q_{j-1,n}^k\right)}{q_{nn}^k} \tag{2.3}$$

where $q_{o,n}^k \triangleq 0$ and a_i, $1 \leq i \leq n$ are as in (2.2). For $Q_k \geq 0$ with $q_{nn}^k = 0$, define $C_k = 0$. Also, define for any symmetric $n \times n$ matrix $S = (s_{ij})$

$$F(S) \triangleq \begin{pmatrix} 0 & 0 \\ 0 & \tilde{F}(S) \end{pmatrix}$$

where

$$\tilde{F}(S) = (s'_{ij}), \quad i,j = 1,2,\ldots,n-1$$

$$s'_{ij} = \begin{cases} s_{ij} - \dfrac{s_{in}\, s_{nj}}{s_{nn}}, & s_{nn} \neq 0 \\ \\ s_{ij}, & s_{nn} = 0 \end{cases}$$

Lemma 2.1. Suppose $Q_k = \left(q^k_{ij}\right)$ is symmetric and $q^k_{nn} \neq 0$. Then, for C_k as in (2.3)

$$(A^TQ_kA - C_k) = F(Q_k).$$

Furthermore, if Q_k is nonegative, so is $F(Q_k)$.

Remark 2.2. When Q_k is symmetric and nonegative with $q^k_{nn} = 0$, it is seen that

$$A^TQ_kA = F(Q_k).$$

2.3. Relation to Linear Systems

Since $B = cb^T$, $c^T = (0\ 0 \ldots 0\ 1)$ and $b^T = (b_1 \ldots b_n)$, (2.1) is equivalent to the linear problem

$$x_{k+1} = Ax_k + cv_k \tag{2.4a}$$

$$\min J = \sum_{k=1}^{N-1} x_k^TQx_k + x_N^TQ'x_N \tag{2.4b}$$

where $v_k = (u_k)\ (b^Tx_k)$ (2.5)

Solution to (2.4) is given for all i, $0 \leq i \leq N-1$, by

$$v_i^* = -\left(\frac{c^TQ_iA}{c^TQ_ic}\right) x_i = -\left(\frac{q_i^TA}{q_i^n}\right) x_i, \quad \text{if } q_i^n \neq 0$$

$$= \text{arbitrary}, \qquad \text{if } q_i^n = 0 \tag{2.6}$$

where q_i is the n^{th} column of Q_i, $q_i^T = (q_i^1, \ldots, q_i^n)$ and

$$Q_i = Q + A^TQ_{i+1}A - \frac{1}{q^n_{i+1}} A^TQ_{i+1}\, c\, c^TQ_{i+1}A, \quad \text{if } q^n_{i+1} \neq 0$$

$$= Q + A^TQ_{i+1}A, \qquad \text{if } q^n_{i+1} = 0 \tag{2.7}$$

With the initial condition $Q_{N-1} = Q'$.

Also, the optimal cost function is given by

$$V_{N-i} = x_i^TF(Q_i)x_i, \quad \text{if } q_i^n \neq 0$$

$$= x_i^T(A^TQ_iA)x_i, \quad \text{if } q_i^n = 0 \tag{2.8}$$

Given any x_o, from $\{v_i^*\}$, the solution to (2.4), if $\{u_i\}$ can be constructed from (2.5), (i.e., $v_i^* = 0$, whenever $b^T x_i^* = 0$) then clearly it also represents the optimal sequence $\{u_i^*\}$. It will be shown that if x_o is in some subset Σ_o of R^n (Σ_o in the notation of Section 2.4), then $\exists\ \{v_i^*\}$ from which $\{u_i^*\}$ can be constructed. However, if $x_o \notin \Sigma_o$ optimum does not exist in general for the bilinear system.

2.4. Discussion of Optimality

Suppose a set Σ_{k+1} is given, and the optimal path in some specified collection of paths $P'(x_{k+1})$ is given for each $x_{k+1} \in \Sigma_{k+1}$. Let the corresponding control sequence be $\{u_i^*\}_{i=k+1}^{N-1}$. Let $P(x_k)$ be any path of the system starting from x_k. Define

$$P'(x_k) = \left\{ P(x_k) \,\middle|\, x_{k+1} \in \Sigma_{k+1}, \text{ and } P(x_{k+1}) \in P'(x_{k+1}) \right\}.$$

Let $R_k^n = R^n - H_k$, where $H_k = \left\{ x_k \,\middle|\, b^T x_k = 0 \right\}$.

Define

$$\Sigma_k = \left\{ x_k \,\middle|\, x_k \in R_k^n \text{ and optimum exists in } P'(x_k); \right.$$
$$\left. \text{i.e., optimum path } \in P'(x_k) \right\}$$

It will be seen that the above definitions, in conjunction with the principle of optimality, can be used to generate the optimal path from a subcollection of paths of the system.

For the system (2.1) we have the following result.

Lemma 2.2. Suppose for each $x_{k+1} \in \Sigma_{k+1}$, the optimal path is given in $P'(x_{k+1})$ with the optimal control sequence $\{u_i^*\}_{i=k+1}^{N-1}$ and the optimal cost $V_{N-(k+1)} = x_{k+1}^T \hat{Q}_{k+1} x_{k+1}$. Then, letting $Q_k = Q + \hat{Q}_{k+1}$ we have

a) Let $\Sigma_{k+1} = R_{k+1}^n - \bigcup_{j=1}^{p} \{x_{k+1} | e_{k+1,j}^T x_{k+1} = 0\}$;

i) if $q_k^n \neq 0$, where q_k is the n^{th} column of Q_k,

then for $x_k \in \Sigma_k$

$$u_k^*(x_k) = -\left(q_k^T A x_k \Big/ q_k^n b^T x_k \right)$$

where

$$\Sigma_k = R_k^n - \bigcup_{j=1}^{p+1} E_{k,j};\ E_{k,j} = \{x_k | e_{k,j}^T x_k = 0\},$$

$$e_{k,j}^T = \begin{cases} e_{k+1,j}^T A - e_{k+1,j}^n \dfrac{q_k^T A}{q_k^n}, & j=1,2,\ldots p \\[2ex] b^T A - b_n \dfrac{q_k^T A}{q_k^n}, & j=p+1 \end{cases}$$

$$V_{N-k} = x_k^T \hat{Q}_k x_k, \quad \hat{Q}_k = F(Q_k).$$

ii) if $q_k^n = 0$, for $x_k \varepsilon \Sigma_k$, u_k^* is any u_k s.t. $x_{k+1} \varepsilon \Sigma_{k+1}$

where

$$\begin{aligned}\Sigma_k &= R_k^n - \bigcup_{j\varepsilon I_o(k+1)} \left\{x_k \middle| \; e_{k+1,j}^T A x_k = 0\right\}, \text{ if } b_n \neq 0 \\ &= R_k^n - \bigcup_{j\varepsilon I_o(k+1)} \left\{x_k \middle| \; e_{k+1,j}^T A x_k = 0\right\} \cup \left\{x_k \middle| b^T A x_k = 0\right\}, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } b_n = 0\end{aligned}$$

where

$$I_o(k+1) = \left\{j \middle| \; e_{k+1,j}^n = 0\right\}.$$

$$V_{N-k} = x_k^T \hat{Q}_k x_k, \quad \hat{Q}_k = A^T Q_k A.$$

b) Let $\Sigma_{k+1} = R_{k+1}^n$;

i) if $q_k^n \neq 0$, then for $x_k \varepsilon \Sigma_k$,

$$u_k^*(x_k) = -\left(q_k^T A x_k \middle/ q_k^n b^T x_k \right)$$

where $\Sigma_k = R_k^n - \{x_k | e_{k,1}^T x_k = 0\}$

$$e_{k,1}^T = b^T A - b_n \frac{q_k^T A}{q_k^n}$$

$$V_{N-k} = x_k^T \hat{Q}_k x_k, \quad \hat{Q}_k = F(Q_k).$$

ii) if $q_k^n = 0$, for $x_k \varepsilon \Sigma_k$, $u_k^* =$ any u_k s.t. $x_{k+1} \varepsilon \Sigma_{k+1}$

$$\Sigma_k = R_k^n \quad \text{if } b_n \neq 0$$

$$\Sigma_k = R_k^n - \{x_k | b^T A x_k = 0\} \text{ if } b_n = 0$$

$$V_{N-k} = x_k^T \hat{Q}_k x_k, \quad \hat{Q}_k = A^T Q_k A.$$

c) $\left\{u_i^*\right\}_{i=k}^{N-1}$ is optimum in the collection of paths

$$P'(x_k) = \{P(x_k) \big| \; x_{k+1} \varepsilon \Sigma_{k+1} \text{ and } P(x_{k+1}) \varepsilon P'(x_{k+1})\}.$$

Remark 2.3.

i) For the bilinear system (2.1) suppose we take at $i = N-1$, $\Sigma_{N-1} = R_{N-1}^n$ and for $x_{N-1} \varepsilon \Sigma_{N-1}$, $P'(x_{N-1}) =$ entire collection of paths from x_{N-1}. Then we have,

$$\text{if } q^n_{N-1} \neq 0: \quad u^*_{N-1} = -\left(\frac{q^T_{N-1}Ax_{N-1}}{q^n_{N-1}b^Tx_{N-1}}\right), \quad V_1 = x^T_{N-1}\hat{Q}_{N-1}x_{N-1}$$

$$\hat{Q}_{N-1} = F(Q_{N-1}), \quad Q_{N-1} = Q';$$

if $q^n_{N-1} = 0$: u^*_{N-1} is arbitrary and

$$V_1 = x^T_{N-1}\hat{Q}_{N-1}x_{N-1}, \quad \hat{Q}_{N-1} = A^TQ_{N-1}A.$$

Now, by applying Lemma 2.2, we can construct $\{\Sigma_i\}$, $\{u^*_i\}$ and $\{Q_i\}$ for $0 \leq i \leq N-2$. For $x_o \epsilon \Sigma_o$, the path generated is optimal in $P'(x_o)$.

ii) Consider the Riccati equation (2.7). It is seen that $\frac{1}{q^n_{i+1}} A^TQ_{i+1}cc^TQ_{i+1}A = c_{i+1}$. Hence, from Lemma 2.1, $Q_i = Q + F(Q_{i+1})$ which is independent of A.

It is easily verified that $\{Q_i\}$, $\{V_{N-i}\}$ for the linear system are the same as those for the bilinear system for $x_k \epsilon \Sigma_k$ and furthermore $u^*_i = \left(v^*_i \big/ b^Tx_i{}^*\right)$. Hence for $x_k \epsilon \Sigma_k$, the path of the bilinear system generated as described in (i) above is the same as the optimal path of the linear system. We have for $x_k \epsilon \Sigma_k$, $k \leq i \leq N-1$,

$$u^*_i = -\left(\frac{q^T_iAx_i}{q^n_ib^Tx_i}\right), \quad \text{if } q^n_i \neq 0 \tag{2.9}$$

$$= \text{arbitrary such that } x^*_{i+1} \epsilon \Sigma_{i+1}, \text{ if } q^n_i = 0$$

and Q_i, V_{N-i} as in (2.7) and (2.8).

The optimality of the path of the bilinear system generated (Remark 2.3 (i)) is discussed in the following.

Theorem 2.1. For $x_k \epsilon \Sigma_k$, the path of the bilinear system generated from Lemma 2.2, with $\Sigma_{N-1}=R^n_{N-1}$ and $P'_{N-1}(x_{N-1}) = \{P(x_{N-1})\}$ for $x_{N-1} \epsilon \Sigma_{N-1}$, is the optimal path of the system.

Proof: Follows since the path of the bilinear system is the same as the optimal path of the linear system for $x_k \epsilon \Sigma_k$. (Remark 2.3 (ii)).

The special case when $q^n_i \neq 0 \; \forall \; i$ (for which a set of sufficient conditions is: $q_{nn} \neq 0$ and $q'_{nn} \neq 0$) is discussed below. The results are specializations of Remark 2.3 (i). The optimal control is unique at each i, for $x_o \epsilon \Sigma_o$. Hence the optimal path is a regular path. Σ_o is the set of initial states for which the regular path exists.

Theorem 2.2. The optimal control for $x_i \epsilon \Sigma_i$, $0 \leq i \leq N-1$ is given by

$$u^*_i(x_i) = -\left(\frac{q^T_kAx_i}{q^n_ib^Tx_i}\right)$$

where

$$Q_i = Q + \hat{Q}_{i+1}, \text{ and the optimal return function is}$$

$$V_{N-i} = x_i^T \hat{Q}_i x_i, \quad \hat{Q}_i = F(Q_i)$$

$$\Sigma_i = R_i^n - \Sigma_i'$$

$$\Sigma_i' = \left\{ x_i \,\middle|\, e_{i,j}^T x_i = 0, \ 1 \le j \le N-i-1 \right\}$$

$$e_{i,j}^T = \begin{cases} e_{i+1,j}^T A - e_{i+1,j}^n \dfrac{q_i^T A}{q_i^n}, & 1 \le j \le N-i-2 \\ b^T A - b_n \dfrac{q_i^T A}{q_i^n}, & j = N-i-1 \end{cases}$$

$\left\{u_i^*\right\}_{i=k}^{N-1}$ is the optimal control sequence in $\{P(x_k)\}$.

Remark 2.4. It is easily shown that across the hyperplanes H_k the optimal cost function is in general discontinuous. For $x_o \notin \Sigma_o$ the optimum may not exist due to the existence of discontinuity.

III. STOCHASTIC OPTIMAL CONTROL

This section extends the study of optimal control to the case when the bilinear system is under the influence of additive and multiplicative noise. The expression for optimal control turns out to be similar in form to that in the deterministic case. It is observed that the presence of noise actually simplifies the analysis compared to that in the deterministic case.

3.1. Problem Statement

Consider the stochastic discrete bilinear system

$$x_{k+1} = (A+u_k B)x_k + H\xi_k + G(x_k)\alpha_k, \quad 0 \le k \le N-1 \tag{3.1}$$

where $x_k \in R^n$, $u_k \in R$, A and B are as in (2.2), H is an $(n \times d_1)$ matrix, and $G(x_k) = \sum_{i=1}^{n} x_k^i G_i$, where G_i, $1 \le i \le n$ are $(n \times d_2)$ matrices. The initial state x_o is Gaussian with $E(x_o) = m_o$, $\mathrm{cov}(x_o) = \Omega_o$. The noise sequences $\{\xi_k\colon\ k=0,1,\ldots,N-1\}$, $\{\alpha_k\colon\ k=0,1,\ldots,N-1\}$ where ξ_k and α_k are vectors of dimension d_1 and d_2 respectively are each Gaussian white and independent of each other and of x_o with mean and covariance

$$E(\xi_k) = 0, \quad E(\xi_k \xi_j^T) = \Xi_k \delta_{kj}$$

$$E(\alpha_k) = 0, \quad E(\alpha_k \alpha_j^T) = \Lambda_k \delta_{kj}$$

respectively, where δ_{kj} is the Kronecker delta function. Exact measurement of the state is assumed available at each i, $0 \le i \le N-1$. The problem is to choose the control policy $\{u_i(x_i)\colon\ 0 \le i \le N-1\}$ so that the (unconditional) cost functional

$$J = E\left\{\sum_{i=1}^{N-1} x_i^T Q x_i + x_N^T Q' x_N\right\}$$

is a min, where Q and Q' are symmetric and nonnegative.

Remark 3.1. Relation to Linear System. Consider the BLS (3.1) and the corresponding LS

$$x_{k+1} = Ax_k + cV_k + H\xi_k + G(x_k)\alpha_k \tag{3.2}$$

$$u_k = \frac{v_k}{b^T x_k} \quad , \quad \text{J as above.}$$

Solution to the linear system (3.2) represents the solution to the bilinear system only if $b^T x_i \neq 0 \; \forall \; i$. Since in any particular realization of the paths, $b^T x_i = 0$ can occur, the solution to the bilinear system needs to be examined. The study also brings out certain features of the stochastic problem, which result in a simplification in analysis over the deterministic problem.

3.2. Optimal Control Policy

Before presenting the main theorem of this section, we give the following result.

Lemma 3.1. For x_k as in (3.1), the conditional expectation of $x_k^T \tilde{g}_k(x_k) x_k$ conditioned on x_{k-1} is $E\left\{x_k^T \tilde{g}_k(x_k) x_k \middle| x_{k-1}\right\} = E\left\{x_k \tilde{C}_k x_k \middle| x_{k-1}\right\}$

where

$$\tilde{g}_k(x_k) = \begin{cases} \tilde{C}_k, & \text{if } b^T x_k \neq 0 \\ 0 & \text{if } b^T x_k = 0 \end{cases} \tag{3.3}$$

Theorem 3.1. The solution to the problem stated in Section 3.1 is given by

$$\underset{0 \le i \le N-1}{u_i^*} = \begin{cases} -\dfrac{\tilde{q}_i^T A^T x_i}{\tilde{q}_i^n b^T x_i}, & \text{if } \tilde{q}_i^n b^T x_i \neq 0 \\ \text{arbitrary, if } \tilde{q}_i^n b^T x_i = 0 \end{cases} \tag{3.4}$$

where $\tilde{q}_i$ is the n^{th} column of $\tilde{Q}_i$ and

$$\tilde{Q}_i = Q + F(\tilde{Q}_{i+1}) + \tilde{G}_{i+1}, \quad 0 \le i \le N-2 \tag{3.5}$$

with initial condition $\tilde{Q}_{N-1} = Q'$,

$$(\tilde{G}_{i+1})_{lj} = \mathrm{tr}(\Lambda_{i+1} G_l^T \tilde{Q}_{i+1} G_j), \quad 0 \le i \le N-2 \tag{3.6}$$

and F(•) is as in Section 2.2. $\tilde{G}_i$ and $\tilde{Q}_i$ are symmetric and nonnegative, $0 \le i \le N-1$. The (conditional) optimal cost for the last (N-i) stages given x_i is given by

$$V_{N-i} = x_i^T\left\{(A^T \tilde{Q}_i A - \tilde{g}_i) + \tilde{G}_i\right\} x_i + \mathrm{tr}\sum_{s=i}^{N-1} (H^T \tilde{Q}_s H)\Xi_s \quad , \quad 0 \le i \le N-1 \tag{3.7}$$

where $\tilde{g}_i$ is as in (3.3), $\tilde{C}_i$ and $\tilde{Q}_i$ are related as in (2.3). Minimum value of J is given by

$$J_{\min} = m_o^T S_o m_o + \mathrm{tr}(S_o \Omega_o) + \mathrm{tr} \sum_{i=0}^{N-1} (H^T \tilde{Q}_i H)\Xi_i \tag{3.8}$$

where

$$S_o = (A^T \tilde{Q}_o A - \tilde{g}_o) + \tilde{G}_o$$

Remark 3.2. In the deterministic problem, we have constructed the set Σ_o which is s.t. if $x_o \in \Sigma_o$, then optimum exists for the BLS. Also if $x_o \notin \Sigma_o$, optimum may not exist as seen in Remark 2.4. This is essentially due to the optimal return function being discontinuous. However, in the stochastic case while constructing the optimal policy via the functional equation, at any k, the initial states can be located anywhere in the space and the singularities determined by $b^T x_k = 0$ do not matter. This is essentially because, the process of averaging which involves integration smoothes out the discontinuities caused by singularities which lie on a set of measure zero. The optimal policy is generated in one stroke without having to deal with the cases $x_o \in \Sigma_o$, as in the deterministic problem. It is easily verified that the optimal policy can be constructed even for those x_o for which optimum does not exist in the deterministic case.

When additive noise is present, the expression for control and the recursive equation for $\tilde{Q}_i$ are similar to those in the deterministic problem when $x_o \in \Sigma_o$. Multiplicative noise affects the recursion equation by contributing an additive term. In both the cases, there is an increase in cost.

VI. CONCLUSIONS

The subject matter of this paper is the optimal control of a class of single-input discrete bilinear systems. Solutions have been obtained for both the deterministic and stochastic problem via the method of dynamic programming.

In the deterministic problem, a procedure has been developed to construct the set Σ_k and the collection of paths $P'(x_k)$ which is s.t. for $x_k \in \Sigma_k$, optimum in $P'(x_k)$ is the same as the optimal path of the bilinear system. Furthermore, this is the same as the optimal path of the associated linear system.

It is noted that the optimal cost function is in general discontinuous across the hyperplane H_k for each k, and it is because of this that for $x_k \notin \Sigma_k$, optimum does not in general exist.

In the stochastic control problem, when additive noise entering directly through a constant gain is present, the expression for control and the recursion relation are the same as in the deterministic case when $x_o \in \Sigma_o$. If noise enters through a linearly state dependent gain, then it directly contributes a term to the recursion equation. The noise, however, does increase the cost in both the cases. In contrast to the deterministic problem, in the stochastic case, the effects of discontinuities in the optimal return function which are confined to sets of measure zero, are wiped out by taking expectations. It is seen qualitatively that the optimal sequence can be found for the stochastic porblem for x_o even when optimum does not exist for the deterministic problem. The effects of discontinuities of the optimal return function which are confined to sets of measure zero propagate in the deterministic case, while they are annihilated in the presence of noise.

ACKNOWLEDGEMENTS

This work was supported partially by the National Science Foundation Grant Nos. ENG 75-09755 and ENG 76-80365.

REFERENCES

1. Mohler, R. R., "Bilinear Structures and Man," in Theory and Applications of Variable Structure Systems (Mohler, R. R., and Ruberti, A., ed.), Academic Press, New York, 1972, pp. 169-181.

2. Goka, T., Tarn, T. J., and Zaborszky, J., "On the Controllability of a Class of Discrete Bilinear Systems," Automatica, September 1973.

3. Tarn, T. J., Elliott, D. L., and Goka, T., "Controllability of Discrete Bilinear Systems with Bounded Control, "IEEE Trans. on Automatic Control, Vol. AC-18, June 73, pp. 298-301.

4. Rink, R. E. and Mohler, R. R., "Completely Controllable Bilinear Systems," SIAM J. Control, Vol. 6, No. 3, pp. 477-486, 1968.

5. DiPillo, G., Bruni, C., and Koch, G., "Bilinear Systems: An Appealing Class of "Nearly Linear" Systems in Theory and Applications," IEEE Trans. on Automatic Control, Vol. AC-19, August 1976, pp. 334-348.

6. D'Alessandro, P., "Structural Properties of Bilinear Discrete-Time Systems," Ric. Automatica, Vol. 3, Aug. 1972.

7. A. Isidori and A. Ruberti, "Realization Theory of Bilinear Systems," in Geometric Methods in System Theory. Edited by D. Q. Mayne and R. W. Brockett, D. Reidel, Dordrecht, Holland, 1973, pp. 83-130.

8. P. D'Alessandro, A. Isidori and A. Ruberti, "Realization and Structure Theory of Bilinear Dynamical Systems," SIAM J. Control, Vol. 12, No. 3, August 1974, pp. 517-535.

9. A. Isidori, "Direct Construction of Minimal Bilinear Realizations from Nonlinear Input-Output Maps," IEEE Trans. on Automatic Control, Vol. AC-18, No. 6, December 1973, pp. 626-631.

10. R. Brockett, "On the Algebraic Structure of Bilinear System," in Theory and Application of Variable Structure Systems. Edited by R. Mohler and A. Ruberti, Academic Press, New York, 1972, pp. 153-158.

APPLICATION OF MULTILINEAR ALGEBRA IN OPTIMAL REGULATION OF NONLINEAR POLYNOMIAL SYSTEMS

M. R. BURIC

Bell Laboratories
Murray Hill, New Jersey 07974

Abstract

The problem of optimal state feedback regulation of polynomial nonlinear systems is considered. The synthesis of optimal regulators is carried out by means of tensor algebraic operations involving symmetric tensor coefficients which appear in system and performance index description. Both time varying and time invariant systems are considered on finite and infinite regulation interval. The algorithms are suitable for computer implementation.

1. Introduction

Optimal nonlinear feedback regulation problem for analytic time invariant systems was conceptually solved by D. L. Lukes in [1].The class of systems treated in his work was given by

$$\dot{x} = F(x,u) = Ax + Bu + f(x,u) \ , \ x \in R^n \ ,u \in R^m \tag{1.1}$$

where F(x,u) is real analytic function in some neighborhood of the origin of R^{n+m} so that f(x,u) is real convergent power series about the origin, beginning with second order terms. The performance criterion was given by

$$J = \int_0^\infty G(x,u)\,dt \tag{1.2}$$

$$G(x,u) = x^TQx + u^TRu + 2x^TCu + g(x,u)$$

$$\begin{bmatrix} Q & C \\ C^T & R \end{bmatrix} > 0 \ .$$

where G(x,u) is also real analytic in the same neighborhood of the origin, so that g(x,u) is convergent power series beginning with third order terms. With the assumption that the pair (A,B) is stabilizable it was shown that the optimal regulator function has the form

$$u(x) = Kx + h(x) \tag{1.3}$$

where h(x) is real convergent power series. In such a case the matrix (A + BK) is a stability matrix, and there is a neighborhood of the origin of R^n where the closed loop system is asymptotically stable. Very importantly, it was shown that the power series h(x) could be actually constructed by solving the nonlinear Hamiltonian equations.

However, the implementation of this result in synthesis of feedback regulators proved to be impossible task for all but elementary low order systems. The work presented by Nishikawa et al. in [2] was based on perturbation techniques, and it was shown that for polynomial nonlinearities the successive terms in feedback regulators could be obtained by iteratively solving certain ordinary linear differential equations. The notion of tensor operations was partially used for regulator synthesis. The systems under consideration were limited to those where control function entered the system equation linearly, and performance index was chosen to be quadratic in state and control variables.

In this paper some of the results given in [4] are summarized, where it was shown that tensor algebra is a natural framework for analyzing polynomial systems. By representing the system equation and performance index as members of space of polynomials in vector variables x(t) and u(t), it is possible to exploit the fact that coefficients of these polynomials belong to the space of mixed tensors. The synthesis of optimal regulators reduces to successive solutions of linear tensor differential equations in case of time varying systems, and linear tensor algebraic equations in case of time invariant systems. Some concepts of tensor algebra and notation used here are outlined in the Appendix.

2. System Description

The class of systems considered is represented by

$$\dot{x}(t) = P(x(t),u(t),t)\ ,\ x(t_0)=x_0\ ,\ x\in R^n\ ,\ u\in R^m\ ,\ t\in[t_0,t_1] \tag{2.1}$$

where the nonlinearity is given by

$$P(x(t),u(t),t) = \sum_{i,j} A^1_{[i],[j]} \overset{i+j}{\circledast} u^{[i]}\otimes x^{[j]}\ ,\ i,j=0,1,2,\cdots \tag{2.2}$$

where

$$A^1_{0,0} = 0\ ,\ A^1_{0,1}(t) = A(t)\ ,\ A^1_{1,0}(t) = B(t)$$

$$A^1_{[i],[j]}(t) : \left[R^m\right]^i \otimes \left[R^n\right]^j \rightarrow R^n$$

The pair of matrices A and B characterizes the linear part of the system,and the tensors $A^1_{[i],[j]}$ completely specify the polynomial or power series in x and u. They are generally time varying, which means that some or all of their components are time varying. Each of them maps the indicated tensor product spaces of R^m and R^n into R^n.

The performance criterion is specified on a finite time interval by

$$J = M(x_f) + \int_{t_0}^{t_1} L(x(t),u(t),t)dt\ \ ,\ x_f = x(t_1)$$

where

$$L(x,u,t) = \sum_{p,q} Q_{[p],[q]} \overset{p+q}{\circledast} u^{[p]}\otimes x^{[q]}\ ,\ p+q \geqslant 2 \tag{2.3}$$

$$M(x_f) = \sum_{k} M_{[k]} \overset{k}{\circledast} x_f^{[k]}\ ,k = 2,3,\cdots$$

The Q tensors are generally time varying , of (0,p+q)-th degree, and M tensors are constant of (0,k)-th degree. They completely characterize the polynomials ,or power series, in vector variables x and u in the performance index expression, and they are mappings defined by

$$Q_{[p],[q]} : \left[R^m\right]^p \otimes \left[R^n\right]^q \rightarrow R,\ \text{and}\ M_{[k]} : \left[R^n\right]^k \rightarrow R$$

2.1. Tensor Valued Polynomial Algebra

Before presenting the results for the regulation problem, some ingredients of the algebra of multivariable polynomials will be given, [3],[4].

Let V be an n-dimensional vector space, and let $v\in V$. Then, a (p,q) tensor valued polynomial of the m-th degree in the n components of v is given by

$$P^p_q(v) = \sum_{i=0}^{m} A^p_{q+[i]} \overset{i}{\circledast} v^{[i]} \tag{2.4}$$

The A coefficients are (p,q+i) tensors, symmetric in the last i indices, and they represent mappings from the i-th degree tensor product space of V into $T^p_q(V)$,the space of (p,q) tensors.

The addition operation is defined among all (p,q) valued polynomials of the same degree by the pointwise addition of their coefficients,i.e.

$$\sum_{i=0}^{m} A^{p}_{q+[i]} \overset{i}{\circledast} v^{[i]} + \sum_{i=0}^{m} B^{p}_{q+[i]} \overset{i}{\circledast} v^{[i]} = \sum_{i=0}^{m} \left[A^{p}_{q+[i]} + B^{p}_{q+[i]}\right] \overset{i}{\circledast} v^{[i]} \tag{2.5}$$

The product of two polynomials, (p,q) and (r,s) tensor valued,results in a (p+r,q+s) tensor valued polynomial whose degree is equal to the sum of degrees of participating polynomials. First, let two polynomials be homogeneous of degrees i and j,then it is shown in [4] that their product satisfies the following equation

$$A^{p}_{q+[i]} \overset{i}{\circledast} v^{[i]} \otimes B^{r}_{s+[j]} \overset{j}{\circledast} v^{[j]} = S_{[i+j]}\Big[\, (i,s)\ (\, A^{p}_{q+[i]} \otimes B^{r}_{s+[j]}\,)\Big] \overset{i+j}{\circledast} v^{[i+j]} \tag{2.6}$$

The operation (i,s) is a permutation operator which acts on the covariant indices of the tensor which results from the indicated product of the two coefficient tensors. It is represented by the following permutation diagram

$$(i,s) = \begin{bmatrix} q & i & s & j \\ q & s & i & j \end{bmatrix}$$

If a sequence of m integers taking values 1,2, ... ,n is denoted by G_n^m,then each entry in the permutation matrix represents a sequence which belongs to the corresponding set, i.e. G_n^q, G_n^i, G_n^s, and G_n^j. Then the interpretation of the permutation (i,s) is that two groups of indices are permuted in the resulting tensor. In addition,the last (i+j) indices are symmetrized by the symmetrization operator $S_{[i+j]}$, which is the operation required by the term $v^{[i+j]}$.

The product of two arbitrary polynomials is derived easily when the above is known, the expression for the tensor coefficient of the resulting polynomial associated with the t-th tensor power of v is given by

$$C^{p+r}_{q+s+[t]} = \sum_{i+j=t} S_{[i+j]}\Big[\, (i,s)\ (\, A^{p}_{q+[i]} \otimes B^{r}_{s+[j]}\,)\Big] \tag{2.7}$$

Additional tools that are required for the regulation problem solution are related to derivatives of tensor valued polynomials in vector variables. It is shown in [4] that the derivative of a homogeneous polynomial at some point v is given by

$$D_v \left[A^{p}_{q+[m]} \overset{m}{\circledast} v^{[m]}\right] = m\ A^{p}_{q+[m]} \overset{m-1}{\circledast} v^{[m-1]} \tag{2.8}$$

It is assumed that the tensor components are constant. The result is a tensor which represents a linear mapping from V into $T^p_q(V)$ at the fixed point v in V where the derivative is taken. By induction it follows that the m-th derivative of the above homogeneous polynomial is equal to

$$m!\ A^{p}_{q+[m]} \tag{2.9}$$

Next, the derivatives of a nonhomogeneous polynomial are obtained easily from the above expression as sums of derivatives of homogeneous terms.

If a polynomial in two vector variables which belong to different vector spaces is considered, then the total derivative at some point (v,x) is given by

$$\begin{aligned} D\ A^{p}_{q+[m],k} \overset{m+k}{\circledast} v^{[m]}\otimes x^{[k]} &= m \left[\, A^{p}_{q+[m],[k]} \overset{m+k-1}{\circledast} v^{[m-1]}\otimes x^{[k]} \right] \circledast dv \\ &+ k \left[\, (m,k)\ A^{p}_{q+[m],[k]} \overset{m+k-1}{\circledast} x^{[k-1]} \otimes v^{[m]} \right] \circledast dx \end{aligned} \tag{2.10}$$

If in a polynomial the vector variable v is function of another scalar variable t, then the derivative of the polynomial with respect to t is given as usually

$$\frac{d}{dt}P(v(t)) = D_v\ P(v(t_0)) \circledast \frac{dv}{dt}\ , \tag{2.11}$$

evaluated at some point $t = t_0$, where (2.8) is used in the first term.

Finally, the concept of raising and lowering indices of a tensor, as it is called in the classical tensor analysis, should be mentioned. It has a natural interpretation in the notation used in this paper. Namely, if the following holds at every point v in V,

$$A^{0}_{[m]} \overset{m}{\circledast} v^{[m]} = v^{*}_{[k]} \overset{k}{\circledast} A^{k}_{[m-k]} \overset{m-k}{\circledast} v^{[m-k]}$$

where v^* is the dual vector of v which belongs to the dual space V^* of V, then it is said that k indices of the tensor were raised.

3. Optimal Regulation Results

The class of nonlinear systems and associated performance indices defined in Section 2. is assumed to represent a model for a local behavior of a control system around some prespecified trajectory maintained by an open loop control function,or around a stationary point. In either case the state and control variables are interpreted as deviations from the reference state trajectory and nominal control function.

In view of this, the initial condition is assumed to belong to an open neighborhood η of the origin of R^n and all assumptions about the properties of the system and performance function are restricted to this neighborhood. The results that are initially obtained for the general case of time varying systems will be specialized for time invariant case.

The tensors which describe the system and performance index are piecewise continuous bounded functions, and L(x,u,t) is assumed to be positive convex function for all $t \in [t_0,t_1]$ in an open neighborhood of the origin of $R^n \times R^m$. Specifically, the quadratic terms of $M(x_f)$ and L(x,u,t) in (2.3) are required to be positive. The correspondence between tensor coefficients with quadratic terms in (2.3) and matrices that are used in the classical linear-quadratic theory as in (1.2) is established by raising appropriate indices

$$Q_{0,[2]} \overset{2}{\circledast} x^{[2]} = x^{*} \overset{1}{\circledast} Q_{0,1}^{\;1} \overset{1}{\circledast} x = x^{*}Qx \quad \textit{which implies} \quad Q_{0,1}^{\;1} = Q$$

Similarly, the matrices R and C are obtained as

$$Q_{[2],0} \overset{2}{\circledast} u^{[2]} = u^{*} \overset{1}{\circledast} Q^{1}_{1,0} \overset{1}{\circledast} u = u^{*} R\, u \quad \rightarrow \quad Q^{1}_{1,0} = R \tag{3.1}$$

$$Q_{1,1} \overset{2}{\circledast} u \otimes x = x^{*} \overset{1}{\circledast} Q_{1,0}^{\;1} \overset{1}{\circledast} u = x^{*}Cu \quad \rightarrow \quad Q_{1,0}^{\;1} = C$$

The positivity of quadratic terms in (2.3) is assumed for all t $\in [t_0,t_1]$ which means that the matrix in (1.2) should be positive definite for all t.

The set of admissible feedback control functions Ω is specified as

$$\Omega = \{\, u(t) : u(t) = \sum_{j} K^{1}_{[j]}(t) \overset{j}{\circledast} x^{[j]} \,\} \tag{3.2}$$

where j=1,2, ... ,and tensor coefficients of this power series are required to be bounded piecewise continuous functions of t,

$$K^{1}_{[j]} : \left[R^n\right]^j \rightarrow R^m$$

These (1,j) tensors, symmetric in all covariant indices, completely specify an optimal analytic feedback controller. The objective is to find a control function in this form for a system given by (2.1) such that for all initial conditions $x_0 \in \eta$ the performance index functional is minimized.

By following the Dynamic Programming approach, the optimal performance function is defined as

$$V(x(t),t) = \min_{u \in \Omega} J(x(t),u(s),t)\,, \quad t_0 \leqslant t \leqslant t_1\,,\ t \leqslant s \leqslant t_1 \tag{3.3}$$

The boundary condition that this function satisfies is

$$V(x(t_1),t_1) = M(x(t_1))$$

In the case considered here the necessary condition for optimality is expressed by the classical functional differential equation in V(x(t),t)

$$\min_{u \in \Omega} \left[\frac{d}{dt} V(x,t)_{u,t} + L(x,u,t) \right] = 0 \tag{3.4}$$

which holds for any initial state x(t) on the optimal trajectory $x_*(t)$.

For the regulation problem defined in this way some constructive results from [4] are summarized.

Existence and Uniqueness There exists an open neighborhood η of the origin, such that if $x_0 \in \eta$, then there exists an optimal control function in Ω. This function is unique if the regulation problem for the truncated linear quadratic problem results in a unique control function. If the regulation interval is infinite the uniqueness is assured by the requirements that the pair (A(t),B(t)) is uniformly completely controllable, and that the pair (A(t),D(t)) is uniformly completely reconstructible, where

$$Q(t) = D^T(t)D(t)$$

Optimal performance function exists for such initial conditions, it is identically equal to 0 if $x_0 = 0$, it is analytic in x for all t, and it has the following form

$$V(x,t) = \sum_k \frac{1}{k!} V_{[k]}(t) \overset{k}{\circledast} x^{[k]} \quad k=2,3,\cdots \tag{3.5}$$

at the optimal trajectory $x(t) = x_*(t)$. The functions $V_{[k]}(t)$ are continuous (0,k) tensors.

Constructive Result The functional differential equation (3.4) is solvable for all unknown tensors in the optimal control and performance functions

$$K^1_{[k]}(t) \text{ and } V_{[k+1]}(t), \quad k=1,2,\cdots$$

In the results that follow the constants $\frac{1}{k!}$ are included in $V_{[k]}$.

The solutions are obtained by substituting the expressions for power series of the optimal performance function V(x(t),t), system equation P(x(t),u(t),t), and L(x(t),u(t),t) into the nonlinear functional equation (3.4) ,which results in

$$\min_u \left[\sum_k \dot{V}_{[k]} \overset{k}{\circledast} x^{[k]} + \sum_k k \left\{ V_{[k]} \overset{k-1}{\circledast} x^{[k-1]} \right\} \circledast \dot{x} \right.$$
$$\left. + \sum_{i,j} Q_{[i],[j]} \overset{i+j}{\circledast} u^{[i]} \otimes x^{[j]} \right] = 0 \tag{3.6}$$

This expression results in two power series equations and the rules of algebra of tensor valued polynomials, which are outlined in Section 2.1. are applicable in solving them for unknown tensor coefficients of V(x,t) and u(x,t) by sorting out all the terms associated with same tensor powers of x(t).

The coefficients are obtained in an iterative procedure, starting with lowest order terms $V_{[2]}$ and K^1_1 first , and then proceeding with other $V_{[k+1]}$ and $K^1_{[k]}$ for k=2,3, ...

The solution for $V_{[2]}$ and K^1_1 turns out to be the classical one obtained in linear-quadratic theory. The term $V_{[2]}$ is identified as the solution V of the matrix Riccati differential equation whose index is lowered, i.e.

$$x^* V x = V_{[2]} \overset{2}{\circledast} x^{[2]} \rightarrow v^i_j = v_{ij} \text{ ,for all } i,j$$

where v^i_j are components of the matrix V, and v_{ij} are the components of $V_{[2]}$. The linear part of the feedback control function is determined by K^1_1 which is the classical m×n matrix. The only elements that participate in this solution are matrices A,B,Q,R,C, and the matrix M^1_1 which defines the boundary condition for the solution matrix V(t), and which is obtained by raising the index of $M_{[2]}$.

The significant result is the solution for other tensor coefficients for k=2,3 ... . The most important conclusion is that the terms $V_{[k+1]}$ and $K_{[k]}$ are given by

$$\dot{V}^{[k+1]} + \left[(k+1)\otimes A_*^T\right] \overset{k+1}{\circledast} V^{[k+1]} + F^{[k+1]} = 0 \tag{3.7}$$

$$\textit{with boundary condition} \quad V^{[k+1]}(t_1) = M^{[k+1]} \quad k=2,3,\cdots$$

and

$$K^1_{[k]} = -\frac{1}{2} R^{-1} \overset{1}{\circledast} T^1_{[k]} \quad ,k=2,3,\cdots \tag{3.8}$$

In (3.7) the matrix $A_* = A + BK^1_1$ is the linear part of the closed loop system, and the meaning of $(k+1)\otimes A_*$ is given by

$$(k+1)\otimes A_* = A_* \oplus A_* \oplus \cdots \oplus A_* \quad , (k+1) \textit{ times}.$$

The operator $\oplus$ is the Kronecker sum of matrices defined as

$$A \oplus B = A \otimes I_m + I_n \otimes B$$

for an n×n matrix A and m×m matrix B, where I_n and I_m are identity matrices of corresponding dimensions.

The equation (3.7) is a linear differential equation in the unknown (k+1,0) tensor variable $V^{[k+1]}$. The (k+1)-fold Kronecker sum of A_*^T is a (k+1,k+1) tensor having very specific relationship with A_*^T, which will be mentioned later. Finally, the term $F^{[k+1]}$ is a (k+1,0) tensor, generally time varying, which has a very important property that it is completely determined from

$$V^{[2]}, V^{[3]}, \cdots, V^{[k]}, \textit{ and } K^1_1, K^1_{[2]}, \cdots, K^1_{[k-1]}, \quad \textit{for } k=2,3\cdots \tag{3.9}$$

by means of well defined tensor algebraic equation.

The expression (3.8) for $K^1_{[k]}$ is algebraic, with another important property that the (1,k) tensor $T^1_{[k]}$ is completely specified by the same set of tensor coefficients as above. This is the essence of the result on solvability of the functional equation , conjectured in [1] and derived in [4] for the time varying case. The equations for $F^{[k+1]}$ and $K^1_{[k]}$ need more space for explanation than it is available here and hence will not be reproduced. It is sufficient to mention that they are uniquely determined by participating tensors (3.9).

The time invariant case and infinite regulation interval is a corollary of the results outlined above. The tensor coefficients of the power series for V(x) and u(x) are time invariant, and the lowest order terms are identical to the linear-quadratic case. The other coefficients are obtained from the following algebraic tensor equations

$$\left((k+1) \otimes A_*^T \right) \overset{k+1}{\circledast} V^{[k+1]} = -F^{[k+1]} \tag{3.10}$$

and (3.8) in which all components are time invariant. The equation (3.10) is linear and can be solved by inverting the matrix representation of the (k+1)-fold Kronecker sum of A_*^T. This inverse exists since the matrix A_*^T is a stability matrix under the assumption that the pair (A,B) is controllable, and since the Kronecker sum of matrices results in a matrix whose eigenvalues are equal to all possible sums of eigenvalues of participating matrices. Therefore, the eigenvalues of the matrix in (3.10) are in the open left half plane of the complex plane which implies its invertibility.

Conclusion. Tensor algebra is an important tool in analyzing polynomial nonlinear systems. By means of algebraic operations outlined in this paper it is possible to exploit inherent algebraic properties of multivariable polynomials. Other than the regulation problem discussed in this paper the same methodology can be applied to estimation questions. It is also likely that some answers related to stability, controllability,and observability can be obtained from studying tensors that describe a polynomial system.

4. Appendix

4.1. Tensor Product Spaces

Let $V_1, V_2, \cdots, V_m$, and P be vector spaces, and $M(V_1, \cdots, V_m:P)$ be the space of all multilinear mappings from $V_1 \times \cdots \times V_m$ into P. Then a pair (P,⊗) is a tensor product of indicated vector spaces if the following holds

(i) $\otimes \in M(V_1, \cdots, V_m:P)$ and all vectors generated by ⊗ form a subspace of P which is equal to P, i.e. Im ⊗ = P.

(ii) If U is any vector space over R and a multilinear mapping $\psi \in M(V_1, \cdots, V_m:U)$ is arbitrary, then a linear mapping $\lambda \in M(P:U)$ exists such that $\psi = \lambda \otimes$.

This decomposition can be shown by the following commutative diagram

$$\begin{tikzcd} V_1 \times \cdots \times V_m \arrow[r, "\otimes"] \arrow[dr, "\psi"'] & P \arrow[d, "\lambda"] \\ & U \end{tikzcd} \tag{A.1}$$

It is shown in [3] that tensor product always exists, and that it is unique up to an isomorphism. Namely, if $(P,\otimes)$ and (Q,μ) are two tensor products of the same collection of vector spaces, then there exists exactly one linear bijective mapping α from P to Q such that

$$\alpha \otimes = \mu$$

The space P will be denoted by

$$V_1 \otimes V_2 \otimes \cdots \otimes V_m \tag{A.2}$$

and will be called tensor product space of indicated vector spaces. An individual element of P will be denoted by $v_1 \otimes \cdots \otimes v_m$. If spaces V_1 *through* V_m are finite dimensional the dimension of P is equal to the product of dimensions of participating spaces. Let bases of $V_1 \cdots V_m$ be $\{e_1\}, \ldots, \{e_m\}$, then the basis of P is composed of all tensor products of basis elements of given bases,i.e.

$$e_{1_{i_1}} \otimes e_{2_{i_2}} \otimes \cdots e_{m_{i_m}}$$

$$i_k \in [1, n_k] \ , k = 1, \cdots, m$$

Any element of P can be represented by a linear combination of its basis elements.

4.2. Mixed Tensors

Let V be an n-dimensional vector space and V^* its dual space. Consider all multilinear mappings μ such that

$$\mu : \underbrace{V^* \times \cdots \times V^*}_{p} \times \underbrace{V \times \cdots \times V}_{q} \rightarrow R$$

Such (p+q)-linear mapping is called a tensor of (p,q)-th degree, p being its contravariant degree, and q being the covariant degree. It will be denoted by t^p_q . The space of all (p,q) tensors is finite dimensional of the dimension n^{p+q}, and it is denoted by $T^p_q(V)$. In [4] it is shown that $T^p_q(V)$ is essentially equal to

$$V \otimes V \otimes \cdots \otimes V \otimes V^* \otimes V^* \otimes \cdots \otimes V^* = [V]^p \otimes [V^*]^q$$

A (p,0) tensor is called p-contravariant, (0,q) tensor is q-covariant, and a (0,0) tensor is a scalar in R.

4.3. Multiplication of Tensors

Let tensors ϕ_q^p and ψ_s^r be given. Their product is a bilinear mapping defined as

$$\otimes : \ T_q^p(V) \times T_s^r(V) \ \longrightarrow \ T_{q+s}^{p+r}(V) \tag{A.3}$$

such that the resulting tensor is given by

$$t_{q+s}^{p+r} = \phi_q^p \otimes \psi_s^r \tag{A.4}$$

For example, let p=q=r=s=1, and let a basis for V be $\{e_i\}$, i=1,...,n, let its dual basis be $\{f_j\}$, j=1,...,n,then the product is

$$\begin{aligned} \phi_1^1 \otimes \psi_1^1 &= \sum a_j^i \, e_i \otimes f_j \ \otimes \ \sum b_l^k \, e_k \otimes f_l \\ &= \sum a_j^i b_l^k \, e_i \otimes e_k \otimes f_j \otimes f_l \end{aligned}$$

The product is distributive over addition,associative,but not generally commutative. The product of a scalar and a tensor is just special case of product of two tensors, since a scalar is (0,0) tensor.

4.4. Contractions of Tensors

Let $T_q^p(V)$ be the space of (p,q) tensors, V being an n-dimensional vector space. A linear mapping

$$C_j^i : \ T_q^p(V) \ \rightarrow \ T_{q-1}^{p-1}(V)$$

defined by the scalar product of i-th contravariant and j-th covariant component of a tensor in $T_q^p(V)$ is called an (i,j) contraction. For example, a contraction of a decomposable tensor is given by

$$\begin{aligned} & C_j^i \ (v_1 \otimes \ldots \otimes v_i \otimes \ldots \otimes v_p \otimes v_1^* \otimes \ldots \otimes v_j^* \otimes \ldots \otimes v_q^*) \\ &= < v_j^*, v_i > (\, v_1 \otimes \ldots v_{i-1} \otimes v_{i+1} \otimes \ldots \otimes v_p \otimes v_1^* \otimes \ldots \otimes v_{j-1}^* \otimes v_{j+1}^* \otimes \ldots \otimes v_q^* \,) \end{aligned}$$

A special operation was introduced in [4], called an m-fold contraction of two tensors, and defined by

$$\phi_q^p \overset{m}{\circledast} \psi_s^r = C_{q-m+1}^{p+1} \cdots C_q^{p+1} \, \phi_q^p \otimes \psi_s^r \tag{A.5}$$

The resulting tensor is in $T_{q-m+s}^{p+r-m}(V)$, and this type of contraction is defined if $m \leqslant q$ and $m \leqslant r$. As an example, let

$$\phi_2^1 = \alpha_{jk}^i \, e_i \otimes f_j \otimes f_k \text{ ,and } \psi_1^2 = \beta_n^{lm} e_l \otimes e_m \otimes f_n$$

then,

$$\phi_2^1 \overset{2}{\circledast} \psi_1^2 = \sum \alpha_{jk}^i \beta_n^{jk} \, e_i \otimes f_n$$

where summations are assumed over all indices. It is shown in [4] that any multilinear mapping μ in $M(V_1, \cdots, V_m : U)$ is uniquely determined by a tensor A_m^1,

$$A_m^1 \in U \otimes V_1^* \otimes \cdots \otimes V_m^*$$

such that the following holds

$$\mu(v_1, \cdots, v_m) = A_m^1 \overset{m}{\circledast} (v_1 \otimes \cdots \otimes v_m)$$

where $v_1 \in V_1, \cdots, v_m \in V_m$.

4.5. Symmetric Tensors

A tensor is called symmetric in its i-th and j-th covariant (or contravariant) indices if the multilinear mapping defined by it is independent with respect to permutation of the i-th and j-th variable. This is equivalent to saying that the components of the tensor are the same if the i-th and j-th indices are interchanged, and that this property holds for any basis system of the underlying vector space V. For example,if the following holds

$$A_3^1 \overset{3}{\circledast} (v_1 \otimes v_2 \otimes v_3) = A_3^1 \overset{3}{\circledast} \sigma(v_1 \otimes v_2 \otimes v_3)$$

where σ is any permutation in the group of permutations of three elements, then A_3^1 is symmetric with respect to all covariant indices, i.e.

$$a_{jkl}^i = a_{\sigma(jkl)}^i \quad , \ \sigma \in S_3 \ .$$

The symmetry of such a tensor will be indicated by $A_{[3]}^1$. In actual computations it is possible to exploit the redundancy that exists in symmetric tensors by considering symmetric tensor product spaces whose dimensions are considerably reduced, [3],[4].

5. References

1. D. L. Lukes, "Optimal Regulation of Nonlinear Dynamic Systems", *SIAM J. Control,* Vol. 7, No. 1, pp. 75-100, 1969.

2. Y. Nishikawa, et al., "A Method for Suboptimal Design of Nonlinear Feedback Systems", *Automatica,* Vol. 7, pp. 703-712, 1971.

3. M. Marcus, *Finite Dimensional Multilinear Algebra,* parts I and II, Marcel Dekker, New York, 1975.

4. M. R. Buric, "Optimal Regulation of Nonlinear Systems by Methods of Tensor Algebra", Ph.D. dissertation, Dep. Elec. Eng., University of Minnesota, Jan. 1978.

LOCAL DECOMPOSABILITY AND THE DISTURBANCE DECOUPLING PROBLEM IN NONLINEAR AUTONOMOUS SYSTEMS

S.H. MIKHAIL and W.M. WONHAM
Dept. of Electrical Engineering
University of Toronto
Toronto, Canada M5S 1A4

ABSTRACT

The problem of decomposability of systems of the form $\dot{x} = F(x,u)$, $z = \delta(x)$ into a two level hierarchy is defined and investigated. The concepts of "amenability" and "totally sufficient amenability" (L.S.A.) are introduced, which include linear (A,B)-invariance as a special case. Closure properties of LSA together with the existence and construction of a locally unique optimal decomposition are proved. As an application of this decomposition property, necessary and sufficient conditions are obtained for the local solvability of the disturbance decoupling problem for the system, $\dot{x} = F(x,u) + \sum_{i=1}^{\sigma} E_i(x)q_i$, $z = \delta(x)$.

INTRODUCTION

The objective of this paper is to report on research in progress whose aim is the development of a constructive analysis to the problem of decomposing a system into a two level hierarchy such that the coarser level alone affects the output. Throughout this paper we shall follow the notation of [4].

1. PARTIAL ORDERING OF ORBITS OF FAMILIES OF VECTOR FIELDS

Definition 1

Let $D_1, D_2 \in V(M)$ be two sets of C^∞-vector fields on the C^∞-manifold $M \cdot D_i^*$ is the smallest Lie subalgebra in $V(M)$ that includes D_i $(i=1,2)$. D_1 is defined to be "locally coarser" than D_2 on M $(D_1 > D_2)$ if $\Delta_{D_1^*} \supset \Delta_{D_2^*}$ for all $m \in M$, where Δ_D is the C^∞-distribution spanned by $D \in V(M)$. The smallest D-invariant distribution which contains Δ_D is denoted by P_D, and it has been shown by [4] that $\Delta_D \subseteq \Delta_{D^*} \subseteq P_D$. If $D_1^* > D_2^*$, then P_{D_1} is D_2^*-invariant, and since P_{D_2} is the smallest D_2^*-invariant distribution, then $P_{D_1} \supseteq P_{D_2}$; thus the orbits of D_2 are submanifolds of the orbits of D_1 on M.

---

* This work was supported in part by the National Research Council of Canada, under Grant No. A-7399.

Definition 2

Let $D_1, D_2 \epsilon V(M)$ with corresponding Lie subalgebras $D_1^*, D_2^* \epsilon V(M)$. The "local meet" D° of D_1 and D_2 on M written as $D^\circ = D_1 \wedge D_2$ is the largest Lie subalgebra on V(M) which is a Lie subalgebra of both D_1^* and D_2^*. Obviously $\Delta_{D^\circ} \subset \Delta_{D_1^*} \cap \Delta_{D_2^*}$, and $P_{D_i} \supseteq P_{D^\circ}$ (i=1,2).

Definition 3

Let $D_1, D_2 \epsilon V(M)$ be Lie subalgebras on V(M). Let

$$\tilde{Q} := \{Q \text{ is a Lie subalgebra of } V(M) \mid Q > D_1, Q > D_2\}$$

The "local join" D_0 of D_1 and D_2 written as $D_0 = D_1 \vee D_2$, is defined to be the Lie algebra $D_0 \epsilon \tilde{Q}$ such that for all $Q \epsilon \tilde{Q}$, $Q > D_0$.

Theorem 1: Let $\tilde{Q}$ be as in Def. 3. $\tilde{Q}$ is closed under $\wedge$.

Proof

Let $Q_1, Q_2 \epsilon \tilde{Q}$. Then $Q_1 \wedge Q_2 > D_i$ (i=1,2), and by Def. 2. $Q_1 \wedge Q_2$ is a Lie subalgebra of V(M). Thus $Q_1 \wedge Q_2 \epsilon \tilde{Q}$. #

Theorem 2: D_0 exists, and is unique.

Proof

Let $Q_* := \bigwedge_{Q \epsilon \tilde{Q}} Q$. By Theorem 1, $Q_* > D_i$ (i=1,2). Let $\bar{D} = [D_1, D_2]_*$ closure under Lie brackets of D_1, D_2 (i.e. $D_1 \vee D_2$), then $Q_* > \bar{D}$. But $\bar{D} > D_i$ (i=1,2), then $\bar{D} \epsilon \tilde{Q}$ and $\bar{D} > Q_*$. Thus $\bar{D} = Q_*$ and for every $Q \epsilon \tilde{Q}$, $Q > \bar{D}$. #

Observe that we have defined a lattice structure on $\tilde{Q} \epsilon V(M)$, which in turn induces a global lattice structure on the orbits passing through each point $m \epsilon M$.

2. DECOMPOSABILITY

Given the system

$$\dot{x} = F(x,u), \qquad z = \delta(x) \tag{2.1}$$

where $x \epsilon X$, $u \epsilon U$, $z \epsilon Z$, X, U, Z are n,m,r dimensional C^∞ manifolds, $F(\cdot,\cdot) \epsilon C^\infty(x \times u)$, and $\delta(\cdot)$ is a C^∞ regular map on X. System (2.1) is said to be decomposable on X into a two level hierarchy if there exists a C^∞ map $\phi : X \rightarrow U$ (called a state feedback) and a Lie subalgebra $S \epsilon V(x)$ that is locally finitely generated (LFG) [4,5] on X, such that

$$[F(x,\phi(x)), S] \subset S \tag{2.2a}$$

where $F(x,\phi(x)) \epsilon V(x)$ is the system vector field. In this case we move between orbits of equal rank (K) [4,5] and $P_S = \Delta_S$. Thus locally, there exists a cover $\{U_\alpha\}$ of $\mathcal{X}$ and C^∞ maps $f_\alpha : U_\alpha \to \mathbb{R}^K$ and C^∞ maps $\eta_\alpha : \mathbb{R}^K \to \mathbb{R}^K$

$$Df_\alpha(x)F(x,\phi(x)) = \eta_\alpha(f_\alpha(x)) \qquad x \in U_\alpha \tag{2.2b}$$

where $f_\alpha = f_\beta$, $\eta_\alpha = \eta_\beta$ $\quad x \in U_\alpha \cap U_\beta$.

Definition 4

A locally finitely generated Lie subalgebra $S \in V(\mathcal{X})$ is said to be amenable on $\mathcal{X}$ if there exists a C^∞ map $\phi : \mathcal{X} \to \mathcal{U}$ such that

(a) $[F(x,\phi(x)),S] \subset S$, $\qquad$ (b) $\Delta_S \subset \operatorname{Ker} D\delta$.

Restricted (A,B)-invariance [3], is a special case of the concept of amenability.

For a given LFG Lie subalgebra $S \in V(\mathcal{X})$ such that $\Delta_s \subset \operatorname{Ker} D\delta$, one would like to establish whether there exists a C^∞ map $\phi : \mathcal{X} \to \mathcal{U}$ such that S is amenable. Combining (2.2b) with the implicit function theorem we obtain the local sufficient condition that if there exists an open neighbourhood of (x_0,u_0) in $\mathcal{X} \times \mathcal{U}$ such that

$$D_1[Df_\alpha(x)F(x,u)]\,\Delta_S \subset \operatorname{Im} D_2[Df_\alpha(x)F(x,u)] \qquad (x,u) \in \Omega \tag{2.4}$$

then there exists a C^∞ map $\phi_\alpha : V_\alpha \to W_\alpha$, where $x_0 \in V_\alpha \subset U_\alpha$, and $W_\alpha \subset \mathcal{U}$ such that $D[Df_\alpha(x)F(x,\phi_\alpha(x))]\,\Delta_s = 0$, $x \in V_\alpha$. (Where $D_1 := \left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n}\right)$, $D_2 := \left(\frac{\partial}{\partial u_1}, \ldots, \frac{\partial}{\partial u_m}.\right)$) This is equivalent to the assertion that for every $\mathcal{X} \in \Delta_s$, there exists a C^∞ map $W : \Omega \to T\mathcal{U}$ such that for $(x,u) \in \Omega$

$$D_1[Df_\alpha(x)F(x,u)]\mathcal{X}(x) + D_2[Df_\alpha(x)F(x,u)]W(x,u) = 0 \tag{2.5}$$

The set of all possible $(\mathcal{X},W)$'s form a C^∞-involutive distribution Δ_s spanned by the Lie subalgebra $S \in V(\Omega)$ called the local associated Lie subalgebra of S on Ω. S is said to be locally sufficiently amenable at (x_0,u_0) (LSA) if (2.4) or (2.5) are satisfied and $\Delta_s \subset \operatorname{Ker} D\delta$.

Theorem 3: Let D_1, D_2 be LFG Lie subalgebras on $V(\mathcal{X})$ that are LSA on $\Omega_1, \Omega_2 \in \mathcal{X} \times \mathcal{U}$ respectively. Then $D_1 \vee D_2$ is LFG on $V(\mathcal{X})$ and LSA on $\Omega_1 \cap \Omega_2$.

Proof

Let $\bar{D}_1, \bar{D}_2$ be the locally associate Lie algebras of D_1, D_2 on $V(\Omega_1), V(\Omega_2)$ respectively. Then $\bar{D}_1 \vee \bar{D}_2$ is a Lie algebra on $V(\Omega_1 \cap \Omega_2)$. It is easily shown that $\bar{D}_1 \vee \bar{D}_2$ is the locally associated Lie algebra of $D_1 \vee D_2 \in V(\mathcal{X})$. Thus $D_1 \vee D_2$ is LSA on $\Omega_1 \cap \Omega_2$. That $D_1 \vee D_2$ is LFG is trivial.

We now give a qualitative construction to identify the maximal LSA Lie algebra $S \in V(G)$ at some point (x_0,u_0).

Constructions:

(1) Let $f^\circ(x) = \delta(x)$, $\Delta^\circ = \mathrm{Ker} D_1\delta$.

(2) Select the largest C^∞- involutive distribution $\Delta^1 \subset \Delta^\circ$ such that $D_1[Df^\circ(x)F(x,u)]\Delta^1(x) \subset \mathrm{Im}\, D_2[Df^\circ(x)F(x,u)]$ on $\Omega^1 \in X \times U$ an open neighbourhood of (x_0,u_0).

(3) Find $f^1(x)$ such that $Df^1(x)\Delta^1(x) = 0$, $f^1(x) \in C^\infty(\Gamma^1)$ where $x_0 \in \Gamma \subset X$ is an open neighbourhood.

(4) Select the largest C^∞-involutive distribution $\Delta^2 \subset \Delta^1$ such that $D_1[Df\,(x)F(x,u)]\Delta^2(x) \subset \mathrm{Im}\, D_2[Df\,(x)F(x,u)]$ on $\Omega^2 \in X \times U$, where $\Omega^2 \subset \Omega^1$.

(5) Find $f^2(x)$, such that $Df^2(x)\Delta^1(x) = 0$, $f^2(x) \in C^\infty(\Gamma^2)$, $x_0 \in \Gamma^2 \subset \Gamma^1$.

Repeat the process such that at the μ^{th} step, we select the largest C^∞-involutive distribution $\Delta^{\mu+1} \subset \Delta^\mu$ such that

$$D_1[Df^\mu(x)F(x,u)]\Delta^{\mu+1}(x) \subset \mathrm{Im} D_2[Df^\mu(x)F(x,u)] \text{ on } \Omega^{\mu+1} \subset \Omega^\mu .$$

At some K^{th} step of the algorithm $\Delta^K = \Delta^{K+1} = \Delta^*$ on $\Omega^K = \Omega^{K+1}$ and Δ^* is spanned by the Lie subalgebra $S^* \in V(\Gamma^K)$

Theorem 4: Let $Q \in V(\Gamma^K)$ be a LSA Lie subalgebra at (x_0,u_0) on Ω^K. Then S^* is the coarsest LSA Lie algebra $\in V(\Gamma^K)$ at (x_0,u_0).

Proof

Since $Q \in V(\Gamma^K)$ is a LSA Lie subalgebra, then, there exist $f_Q \in C^\infty(\Gamma^K)$ such that $Df_Q(x)\Delta_Q(x) = 0$ on Γ^K, and where $D_1[Df_Q(x)F(x,u)]\Delta_Q \subset \mathrm{Im} D_2[Df_Q(x)F(x,u)]$ $(x,u) \in \Omega^K \subset \Omega^1$. But $\Delta^\circ \supset \Delta_Q$. If at any stage $\Delta_Q \subset \Delta^\mu$ on $\Gamma^K \subset \Gamma^\mu$, $Df^\mu(x)\Delta_Q = 0$, (i.e. $f^\mu(x) = \pi(f_Q(x))$ on Γ^K), thus

$$D_1[Df^\mu(x)F(x,u)]\Delta_Q \subset \mathrm{Im} D_2[Df^\mu(x)F(x,u)]$$

but by construction 1, $\Delta^{\mu+1} \subset \Delta^\mu$ is the largest C^∞ involutive distribution on $\Gamma^K \subset \Gamma^\mu$, thus $\Delta_Q \subset \Delta^{\mu+1}$. By induction $\Delta_Q \subset \Delta^*$ on Γ^K. Thus $Q < S^*$ on Γ^K.

#

3. A LOCAL DISTURBANCE DECOUPLING PROBLEM (LDDP)

Consider the system (2.1) with disturbances in the form

$$\dot{x} = F(x,u) + \sum_{i=1}^{\sigma} E_i(x)q_i, \qquad z = \delta(x) \tag{3.1}$$

where $E_i(0)\in C^\infty(X)$ and $(q_1,\ldots,q_\sigma)$ are disturbance variables.

The Disturbance Decoupling Problem (DDP) can be stated as: "For system 3.1, find if possible a C^∞ map $\phi: X \to U$ such that the output z is independent of the disturbance q".

If we define the vector fields $\bar{E}_i \in V(X)$, $\bar{E}_i := \sum_{j=1}^{n} E_{i,j}(x)\frac{\partial}{\partial x_j}$ $(i\in\underline{\sigma})$ and E^* is the smallest Lie algebra of $(\bar{E}_i,\ i\in\underline{\sigma})$, then, "DDP is solvable if there exists a C^∞ map $\phi: X \to U$ and a LFG Lie subalgebra $Q\in V(X)$ such that

(i) Q is amenable

(ii) $Q > E^*$.

DDP is said to be locally sufficient solvable (LSS) at (x_0,u_0) in $X\times U$, if there exists Lie subalgebra Q that is LSA on $\Omega\subset\Omega^K$ an open neighbourhood of (x_0,u_0) such that $Q>E^*$ for $x\in\Gamma\subset\Gamma^K$.

Theorem 5: For system (3.1), DDP is locally sufficient solvable if and only if $S^*>E^*$ on $\Gamma\subset\Gamma^K$.

Proof

(If) By definition of LSS of DDP on (x_0,u_0),
(Only if) Let Q be any LSA Lie algebra on $\Omega\subset\Omega^K$ an open neighbourhood of (x_0,u_0) such that $Q>E^*$ for $x\in\Gamma\subset\Gamma^K$. Then by Theorem 4 $S^*>Q>E^*$ on $\Gamma\subset\Gamma^K$.

#

CONCLUSION

Several important questions remain to be answered (i) Can the idea of local sufficient amenability be globalized, and under what conditions would we obtain a unique global maximal amenable distribution. (ii) Can we stabilize the system about the operating point (x_0,u_0) such that we have an open neightbourhood $T^K\subset\Omega^K$ with the property that for any $(x(0),u(0))\subset T^K$, $(x(t),u(t))\subset T^K$ $t\geq 0$. (iii) Is there an easy way to compute the state feedback, since even in the local situation this involves solving a partial differential equation.

References

[1] W.M. Boothby. An Introduction to Differentiable Manifolds and Riemannian Geometry. Academic Press, 1975.
[2] S. Lang. Differentiable Manifolds. Addison-Wesley Pub. Co., 1972.
[3] W.M. Wonham. Linear Multivariable Control: A Geometric Approach. Springer-Verlag, New York, 1974.

[4] H.J. Sussmann. Orbits of Families of Vector Fields and Integrability of Distributions. Trans. Amer. Math. Soc. 180 (1973), pp. 171-188.

[5] R. Hermann. The Differential Geometry of Foliations II, J. Math. and Mech. Vol. II, No. 2 (1962), pp. 303-315.

A DOUBLE INTEGRAL QUADRATIC COST AND TOLERANCE OF FEEDBACK NONLINEARITIES[†]

W. H. Kwon* and A. E. Pearson
Division of Engineering and Lefchetz Center for Dynamical Systems, Brown University, Providence, RI 02912. *Department of Electrical Engineering, Seoul National University, Seoul, Korea.

ABSTRACT

A double integral quadratic cost with an associated integral constraint on state trajectories is shown to result in a stable feedback control law for linear time varying differential systems. The gain matrix for this control is obtained by integrating a Riccati type matrix differential equation over a finite time interval and is shown to allow for a large class of nonlinearities in the feedback loop without destroying its asymptotically stable property.

INTRODUCTION

A particularly simple way to stabilize a time invariant linear control system by means of state variable feedback is the method of Kleinman [1]. Specifically, if the control system is characterized by the state equations

$$\dot{x} = Ax + Bu \tag{1}$$

where (A,B) is assumed to be a controllable pair of nxn and nxm matrices, then the constant coefficient linear feedback control law

$$u = -B'W^{-1}(T)x \tag{2}$$

stabilizes (1) for any $T > 0$, where W^{-1} is the inverse of the controllability Gramian

$$W(T) = \int_0^T e^{-At}BB'e^{-A't}dt. \tag{3}$$

A suitable Lyapunov function for verifying the asymptotic stability of (1) and (2) is

$$V(x) = x'W^{-1}(T)x. \tag{4}$$

If a nonlinearity is inserted in the feedback loop (see Fig. 1) in order to test the tolerance of the resulting control system to such an anomaly, then the control signal is defined by

$$u = \eta(v) \quad , \quad v = -B'W^{-1}(T)x \tag{5}$$

[†] This research was supported in part by the Air Force Office of Scientific Research under grant AFOSR-75-2793.

where $\eta(\cdot)$ is a memoryless function mapping $\mathcal{R}^m$ into $\mathcal{R}^m$ whose slope restrictions are to be examined by stability considerations. Accordingly, the time derivative of $V(x)$ along solution curves of (1) and (5) is found to be

$$\dot{V} = x'W^{-1}(WA' + AW)W^{-1}x - 2v'\eta(v) \tag{6-a}$$

$$= -x'W^{-1}(\frac{dW}{dT} - BB')W^{-1}x - 2v'\eta(v) \tag{6-b}$$

where the last equality follows from the fact that (3) is the solution to the Lyapunov matrix differential equation

$$\frac{dW}{dT} = -AW - WA' + BB' \quad , \quad W(o) = 0 \ . \tag{7}$$

Supposing η to be a bounded nonlinear function such that

$$v'\eta(v) \geq v'v \qquad \text{for all } v \varepsilon \mathcal{R}^m, \tag{8}$$

it is seen that (6-b) in combination with (5) and (8) implies the inequality

$$\dot{V} \leq -x'W^{-1}(\frac{dW}{dT} + BB')W^{-1}x. \tag{9}$$

Asymptotic stability is then assured using arguments analogous to Anderson and Moore [2, Sec. 6.1] relative to (4) and (9).

In the case of a single input system, inequality (8) implies that $\eta(v)$ is restricted to the first and third quadrants and is bounded below by the linear function with unity slope (gain). This is similar to the tolerance for nonlinearities property of the standard optimal steady state linear regulator problem [2,3], although the lower bound on the nonlinearity is less restrictive in the latter situation (i.e. the lower bound is a linear function with gain one half for the standard optimal linear regulator problem verses unity for the control law (2)). However, this lower bound can be made less conservative by recognizing that T in (2) is any positive number and assuming that η satisfies (cf. Eq. (8)):

$$v'\eta(v) \geq \frac{T}{2}\, v'v \quad \text{for all } v \varepsilon \mathcal{R}^m. \tag{10}$$

Then (6-a) in combination with (5) and (10) implies

$$\dot{V} \leq -x'W^{-1}(-AW - WA' + TBB')W^{-1}x \tag{11}$$

which is still negative definite. Comparing (9) and (11) let a new matrix W(T) be defined implicitly by

$$\frac{d\tilde{W}}{dT} = -A\tilde{W} - \tilde{W}A' + TBB' \quad , \quad \tilde{W}(o) = 0 \ . \tag{12}$$

It is clear from (7) that W and $\tilde{W}$ are related by

$$\tilde{W}(T) = \int_o^T W(t)dt \tag{13}$$

and hence $\tilde{W}(T)$ is a symmetric positive definite matrix. Furthermore, if the new control law

$$u = -B'\tilde{W}^{-1}(T)x \tag{14}$$

is used in connection with (1), the resulting feedback system can be similarly shown to be asymptotically stable using as a Lyapunov function

$$V(x) = x'\tilde{W}^{-1}(T)x, \quad 0 < T \leq 2.$$

Notice that in this case the feedback system can tolerate nonlinear functions $\eta(v)$ in the loop which satisfy inequality (10) for any $T > 0$ without destroying asymptotic stability. In fact, for any given nonlinear function lying in the first and third quadrant it is possible to choose a suitably small $T > 0$ so that (10) is satisfied while still retaining asymptotic stability for the control law (14). Hence the control law (14) admits a larger class of nonlinearities in the loop without destroying stability, yet it is essentially as easy to compute as the control law (2).

A DOUBLE INTEGRAL QUADRATIC COST PROBLEM

Another property of Kleinman's control law (2) is that it has been shown to be related to the following optimal control problem for (1):

$$\text{Minimize } J(u) = \int_t^{t+T} u'(\tau)u(\tau)d\tau \tag{15-a}$$

$$\text{Subject to} \quad x(t + T) = 0 \ . \tag{15-b}$$

Thomas and Barraud [4,5] showed that the feedback version of the above receding horizon cost problem subject to the finite horizon constraint (15-b) is identical with the control law (2) in the sense that if the optimal open loop solution, $u^*(\tau,x(t))$, $t \leq \tau \leq t + T$, is made into a feedback control law by placing $\tau = t$, then $u^*(t,x(t))$ reduces to (2). This result was generalized in a variety of ways by the authors in [6].

In relation to the above discussion, the question arises as to what interpretation (if any) can be given to the control law (14) in which $\tilde{W}$ is obtained by a pure integration of W as shown by (13). Standard calculus of variations arguments can be used to establish that (14) arises from the following receding horizon quadratic cost problem:

$$\text{Minimize } J(u) = \int_t^{t+T} \int_t^{\tau} u'(s)u(s)ds d\tau \tag{16-a}$$

$$\text{Subject to } \int_t^{t+T} \int_t^{\tau} e^{A(t-s)}Bu(s)ds d\tau = -x(t). \tag{16-b}$$

Although not particularly obvious, it is straightforward to show that the feedback version of the above double integral quadratic cost problem subject to the double integral finite horizon constraint (16-b) is precisely (14). That is, putting $\tau = t$ into the open loop optimal control $u^*(\tau;x(t))$ for (16) leads directly to the feedback control (14) analogous to the relation between the solution to (15) and the Kleinman control law (2). Since the control law of [6] applies to time varying linear control systems and includes (2) as a special case, it is natural to consider the generalization of (16) along the lines of the formulation in [6] bearing in mind the potential that exists in enlarging the tolerance for nonlinearities in the feedback loop without destroying stability. In addition, there will result a new method for stabilizing linear systems by state variable feedback for which there are very few methods in the case of time varying systems.

TIME VARYING LINEAR SYSTEMS

Consider the linear time varying control system

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) \end{aligned} \tag{17}$$

where $(A(t),B(t),C(t))$ are nxn, nxm and pxn piecewise continuous bounded matrices, with the pairs $(A(t),B(t))$ and $(A(t),C(t))$ assumed to be uniformly completely controllable and observable, respectively. Let a linear time varying feedback control law be defined by

$$u(t,x(t)) = -R^{-1}(t)B'(t)\tilde{P}^{-1}(t,t+T)x(t) \tag{18}$$

where

$$\tilde{P}(t,t+T) = \int_t^{t+T} P(t,\tau)d\tau \ , \quad T > 0 \tag{19}$$

and $P(t,\tau)$ is obtained from the matrix Riccati equation

$$-\frac{\partial}{\partial s}P(s,\tau) = -A(s)P(s,\tau) - P(s,\tau)A'(s) - P(s,\tau)C'(s)Q(s)C(s)P(s,\tau) + B(s)R^{-1}(s)B'(s) \quad , \quad P(\tau,\tau) = 0 \ , \quad s \leq \tau \tag{20}$$

relative to a given pair of symmetric nonnegative definite weighting type matrices $(Q(t),R(t))$, with $R(t) > 0$ for all t. It can be shown by standard calculus of variations arguments that (18) stems from the following double integral quadratic cost problem with associated double integral finite horizon time constraint for the case $Q = 0$:

$$\text{Minimize } J(u) \int_t^{t+T} \int_t^{\tau} u'(s)R(s)u(s)ds d\tau^{\dagger} \tag{21-a}$$

$$\text{Subject to} \quad \int_t^{t+T} \int_t^{\tau} \Phi(t,s)B(s)u(s)ds d\tau = -x(t). \tag{21-b}$$

In the above, $\Phi(t,s)$ is the state transition matrix for $A(t)$ in (17). From the variation constants formula, (21-b) is equivalent to the following integral constraint on the state:

$$x(t) + \int_t^{t+T} \Phi(t,s)x(s)ds = Tx(t) \tag{21-c}$$

Hence the constraint (21-b) implies the terminal constraint $x(t+T) \to 0$ for $T \to 0$, as seen from (21-c). However, the control law (18) is defined only for T strictly positive. The case $Q \neq 0$ allows some weighting on the state analogous to the cost (1.2) in [6].

The symmetric positive definite matrix $P(t,\tau)$ defined by (20) is the same matrix as the $P(\tau,\sigma)$ matrix defined by Eq. (19) in [6]. Since $\tilde{P}(t,t+T)$ is the pure integration of $P(t,\tau)$ over $t \leq \tau \leq t + T$, the matrix $\tilde{P}(t,t+T)$ satisfied boundedness and positive definite properties analogous to those listed in Lemma 2.1 of [6] for $P(t,\tau)$. The following theorem can be established utilizing these properties. In the statement of this theorem, δ_c is the same positive constant as used in [6] for the uniform controllability and observability definitions (see Eqs. (1.11) - (1.16) of [6]).

Theorem 1 Assume there exist positive constants $(\alpha_1,\alpha_2,\alpha_3)$ such that $\alpha_1 I \leq R(t) \leq \alpha_2 I$ and $0 \leq Q(t) \leq \alpha_3 I$. The system (17) with the feedback control (18) is uniformly asymptotically stable for any fixed T satisfying $\delta_c < T \leq 2$. Furthermore, the feedback system remains uniformly asymptotically

---

[†] In showing that (18) stems from (21) the open loop optimal control for (21), $u^*(\tau,x(t))$, $t \leq \tau \leq t + T$, is made into a feedback control by putting $\tau = t$, thus obtaining (18) in the same manner as was discussed in relating (14) and (16), or (2) and (15).

stable with either of the following modifications to the control law (18):

$$u(t) = -\psi(t)R^{-1}(t)B'(t)\tilde{P}^{-1}(t,t+T)x(t) \tag{22}$$

where the mxm time varying matrix $\psi(t)$ is such that

$$\psi'(t)R(t) + R(t)\psi(t) \geq TR(t), \tag{23}$$

or (see Fig. 1)

$$u(t) = \eta(-R^{-1}(t)B'(t)\tilde{P}^{-1}(t,t+T)x(t)) \tag{24}$$

where $\eta(\cdot)$ maps $\mathcal{R}^m$ into $\mathcal{R}^m$ such that

$$v'R(t)\eta(v) \geq \frac{T}{2} v'R(t)v \quad \text{for all } v\varepsilon\mathcal{R}^m. \tag{25}$$

The proof of Theorem 1 (which is omitted here) can be obtained by taking as a Lyapunov function

$$V(x,t) = x'\tilde{P}^{-1}(t,t+T)x,$$

where $\tilde{P}(t,t+T)$ is defined by (19) and (20), and verifying that $\dot{V}$ is negative definite along solution curves of (17) with the control (22) or (24). It is noted that nonlinearities satisfying $\psi(t) \geq T/2$ are allowed in the feedback loop for a single input system and that for a given nonlinear function in the first and third quadrants it is possible to find a suitable value of T so that asymptotic stability of the system (17) and (18) is retained. As in the case of the control presented in [6] for time varying systems, the gain matrix $\tilde{P}(t,t+T)$ in (18) is obtained by integrating the Riccati equation (20) backward in time over a finite time interval for each t and is essentially as easy to obtain as the control law in [6].

TIME INVARIANT LINEAR SYSTEMS

The corresponding result for time invariant control systems can be summarized as follows.

Theorem 2. If the matrices (A,B,C,Q,R) in (17) and (21) are constant, with the pairs (A,B) and (A,C) assumed (as usual) to be controllable and observable respectively, then (17) is uniformly asymptotically stable with the constant coefficient feedback control law

$$u = -R^{-1}B'\tilde{P}^{-1}(T)x \tag{26}$$

where $0 < T \leq 2$ and $\tilde{P}(T)$ is obtained from

$$\tilde{P}(T) = \int_o^T P(t)dt \tag{27}$$

and $P(t)$ is defined by

$$\dot{P}(t) = -AP(t) - P(t)A' - P(t)C'QCP(t) + BR^{-1}B' \tag{28}$$

$$P(o) = 0$$

for any given pair of symmetric nonnegative definite weighting matrices (Q,R) with $R > 0$. Furthermore, the feedback system remains uniformly asymptotically stable with either of the following modified control laws:

$$u(t) = -\psi(t)R^{-1}B'\tilde{P}^{-1}(T)x(t) \tag{29}$$

where $\psi(t)$ satisfies (23) with a constant R matrix, or with the control law

$$u(t) = \eta(-R^{-1}B'\tilde{P}^{-1}(T)x(t)) \tag{30}$$

where η satisfies (25) for some $T\epsilon(0,2]$.

Although the above is a special case of Theorem 1, a direct proof can be given using the Lyapunov function

$$V(x) = x'\tilde{P}^{-1}(T)x.$$

Note that for the special case $Q = 0$, $\tilde{P}(T)$ in (27) can be obtained by a single integration; viz. integrating both sides of (28) with $Q = 0$:

$$\begin{aligned} P(t) = \frac{d\tilde{P}(t)}{dt} &= -A\tilde{P}(t) - \tilde{P}(t)A' + tBR^{-1}B' \\ \tilde{P}(o) &= 0 . \end{aligned} \tag{31}$$

Thus (26) includes (14) as a special case (cf. (12) and (31)).

Some characteristics of the control (26) in the frequency domain will now be considered. For notational convenience we define

$$K \triangleq \tilde{P}^{-1}(T), \; L \triangleq -R^{-1}B'\tilde{P}^{-1}(T),$$

and

$$\tilde{Q} \triangleq \tilde{P}^{-1}(T) \int_o^T P(t)C'QCP(t)dt \; \tilde{P}^{-1}(T) + \tilde{P}^{-1}(T)P(T)\tilde{P}^{-1}(T).$$

Then (28) is equivalent to the following matrix equation:

$$A'K + KA - T\,KBR^{-1}B'K + \tilde{Q} = 0. \tag{32}$$

For simplicity we consider a single input system where R can be taken as unity, B = b is a column vector and L = ℓ is a row vector. By a slight modification of the development given in Sec. 5.2 of [2], (32) can be shown to yield the following relation in the frequency domain:

$$\left| \frac{1}{T} - \ell(j\omega I - A)^{-1}b \right| \geq \frac{1}{T}. \tag{33}$$

Inequality (33), which is analogous to Eq. (5.2 - 10) in [2], implies that the Nyquist plot of the closed-loop transfer function $-\ell(j\omega I - A)^{-1}b$ lies outside a circle centered at $-\frac{1}{T} + jo$ with radius $\frac{1}{T}$ as shown in Fig. 2. The key observation, as noted in [2], is that the number of encirclements of any point inside this circle is the same. Theorem 2 in conjunction with inequality (33) shows that the number of encirclements of points lying on the real axis inside this circle is the same as the number of poles of the transfer function in the closed righthand plane. From these properties and Fig. 2 it is clear that the system employing the control law (26) has an infinite gain margin and a phase margin at least as great as the following:

$$\Theta_m = \tan^{-1}\sqrt{\frac{4}{T^2} - 1}\,.$$

Thus, $\Theta_m = 0$ for T = 2, $\Theta_m = 60°$ for T = 1, and $\Theta_m \to 90°$ as $T \to 0$. From the well known circle criterion the nonlinear function $\psi(t) \geq \frac{T}{2}$ can be allowed in the closed-loop without destroying its stability. This is consistent with Theorem 2.

CONCLUSIONS

It has been recognized that relatively few optimal control problems lead to stable linear feedback control laws, particularly for time-varying linear systems. In this paper an unconventional optimal control problem has been introduced whose modification leads to a stable control. Since the stabilizing feedback gains are obtained by integrating a Riccati equation over a finite time interval, its computation should be easy with modern computers. The control law (26) employing (31) is one of the easiest ways to stabilize a linear time invariant system like the one in [1], since the gain matrix can be obtained by a single integration of a Lyapunov-type equation as indicated in (31). The results indicate that a short receding horizon distance T should be chosen when large nonlinearities exist in the feedback loop. Stabilization with a prescribed degree of stability can be obtained exactly the same as in [6].

REFERENCES

[1] D. L. Kleinman, "An easy way to stabilize a linear constant system", IEEE Trans. on Automat. Contr., Vol. AC-15, p. 692, 1970.

[2] B. D. O. Anderson and J. B. Moore, Linear Optimal Control, Englewood Cliffs, NJ, Prentice-Hall, 1971.

[3] M. G. Safonov and M. Athans, "Gain and phase margin for multiloop LQG regulators", IEEE Trans. on Automat. Contr., Vol. AC-22, p. 173-179, 1977.

[4] Y. Thomas and A. Barraud, "Commande optimale a horizon fuyant", Francaise d'Automat. Inform. et de Recherche Operat., Vol. J-1, p. 126-140, Apr. 1974.

[5] Thomas, Y. A., "Linear quadratic optimal estimation and control with receding horizon", Electronics Letters, Vol. 11, p. 19-21, 9 Jan. 1975.

[6] W. H. Kwon and A. E. Pearson, "A modified quadratic cost problem and feedback stabilization of a linear system", IEEE Trans. on Automat. Contr., Vol. AC-22, p. 838-842, October 1977.

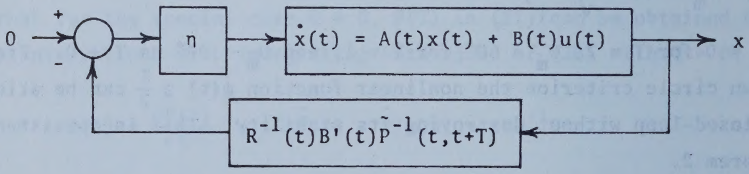

0
+
-
η
$\dot{x}(t) = A(t)x(t) + B(t)u(t)$
x
$R^{-1}(t)B'(t)\tilde{P}^{-1}(t,t+T)$

Fig. 1. Nonlinearity in the Loop

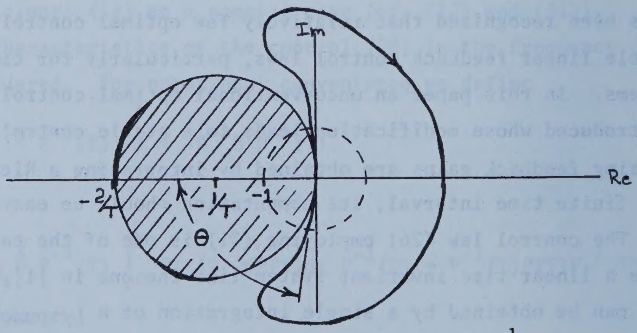


Fig. 2. A Nyquist Plot of $-\ell(j\omega I-A)^{-1}b$

A NEWTON-LYAPUNOV DESIGN FOR A CLASS OF NONLINEAR REGULATOR PROBLEMS[†]

JOE H. CHOW
EUSED, General Electric Company
Schenectady, New York

ABSTRACT

The nonlinear regulator problem considered has the same control structure as the linear regulator problem. Two results from the linear regulator problem are extended. The positive definite solution satisfying the sufficiency condition yields the unique optimal control, and a Newton-Lyapunov procedure iteratively improves feedback controls with a fixed domain of stability. Numerical aspect of the method using a suboptimal scheme is discussed.

1. Introduction

In contrast to the well-developed theory of the linear regulator problem, there are relatively few results on the nonlinear regulator problem [1-10]. The main difficulty lies in solving the Hamilton-Jacobi equation arising in such problems for the optimal feedback control. In [1-5], based on the assumptions that the nonlinearities are weak and the linearized problem is controllable (stabilizable) and observable (detectable), feedback controls are obtained using matched asymptotic expansions. Numerical computation of the series expansions involves tensor equations, and the domain of stability depends on the truncation of the series expansion of the control.

The objective of this paper is to extend some results from linear regulators to a class of nonlinear regulators using the iterative scheme in [10]. In particular, we obtain analogs of the stabilizing solution to the Riccati equation and the Newton-Lyapunov method for computing the Riccati solution in nonlinear regulators. The iterative scheme differs from [1-5] in that it successively generates improving controls while maintaining a fixed domain of stability. Exponential stability which is crucial in [1-5] is not essential here.

We consider a class of nonlinear regulators where the system is linear in the control and the cost function to be minimized is a quadratic form of the control. Due to the structure of the problem, the Hamilton-Jacobi (H-J) equation yields a feedback control law with a simple structure. Our first result shows that the stabilizing solution of the H-J equation is the unique optimal solution. Applied to this problem, the iterative scheme of [10] reduces to a Newton-Lyapunov design which requires an "admissible" control to initiate the algorithm. At each stage of the iteration, we

---

[†] This work was performed at the Decision and Control Laboratory, CSL, University of Illinois, Urbana, Illinois, and was supported in part by the Department of Energy, Electric Energy System Division, under Contract U.S. ERDA EX-76-C-01-2088 and in part by the Air Force under Grants AFOSR-73-2570 and AFOSR-78-3633.

improve the feedback control which possesses a domain of stability not smaller than that of the initiating control. The controls are successively solved for from a system of linear partial differential equations, which is an analog of the matrix Lyapunov equation appearing in the iterative solution of the Riccati equation for linear regulators [11]. Furthermore, the improvement in the cost function is quadratic. The uniqueness result guarantees that if convergence occurs, the design method yields the optimal solution.

The numerical solution to the partial differential equations is computed using the method of characteristics which deals with an equivalent system of ordinary differential equations. The result is a feedback control map. In practice, to reduce the amount of data storage and computation, suboptimal schemes such as polynomial approximations, can be used. We illustrate the method by a second order example.

The organization of the paper is as follows. We formulate the problem in Section 2 and show the uniqueness property in Section 3. The iteration scheme and its properties are discussed in Section 4. Numerical aspects and the example are shown in Section 5.

2. The Nonlinear Regulator Problem

We consider the nonlinear regulator problem with the structure that the control enters linearly in the system

$$\dot{x} = f(x) + B(x)u \qquad (1)$$

where $x \varepsilon R^n$, $u \varepsilon R^m$, and appears quadraticly in the cost function

$$J = \int_0^\infty (p(x) + \tfrac{1}{2} u'R(x)u)dt. \qquad (2)$$

Thus the control structure of this problem is the same as the linear regulators. A particular case of this problem is the bilinear system [8,9]. We assume that f, B, p and R are sufficiently smooth functions of x.

For the well posedness of the problem, we need the following assumptions.

I. $f(0) = 0$.

II. R is a positive definite matrix and $c_1 < \|R\| < c_2$ for some fixed positive c_1 and c_2.

III. p is positive definite with $p(0) = 0$ and $p \to \infty$ as $x \to \infty$.

IV. There exists a sufficiently smooth control $u^{(0)}$ for system (1) such that $x = 0$ is the globally asymptotically stable equilibrium. Furthermore, $u^{(0)}$ is an admissible control in the sense that the resulting cost function J using this control is bounded for all bounded $x(0)$.

Assumption III establishes that the origin $x = 0$ is the desired asymptotically stable equilibrium as it excludes the possibility that J remains finite as $x \to \infty$. Assumption II and the boundedness of J imply that the admissible control $u^{(0)}(x)$ must possess the property $u^{(0)}(0) = 0$.

The admissible control will be used to initiate the proposed iterative design of the control for the nonlinear regulator problem (1), (2). In linear regulators, exponential stability of the designed feedback system guarantees that the quadratic cost function will be bounded. For the nonlinear problem (1), (2), not every asymptotically stabilizing control is admissible. The control $u = -x^2-x^5$ asymptotically stabilizes the system

$$\dot{x} = x^2 + u. \tag{3}$$

However, if the cost function is

$$J = \int_0^\infty (x^4 + \tfrac{1}{2}u^2)dt, \tag{4}$$

then $J = \infty$ for all $x(0) \neq 0$ as

$$x(t) = \mathrm{sign}(x(0))\ (4t + (x(0))^{-4})^{-\frac{1}{4}}. \tag{5}$$

Assumption IV excludes this situation.

We now provide a direct means of determining whether the asymptotically stabilizing control u(x) is admissible or not. Let V(x) be any Lyapunov function for the feedback control system

$$\dot{x} = f(x) + B(x)\ u(x), \quad x(0) = x_0 \tag{6}$$

with $V(0) = 0$ and $\dot{V}(x) = -q(x)$ where $q(x)>0$, $q(0) = 0$.

Lemma 1:

If $q(x) \geq p(x) + u'(x)R(x)u(x)$ in a domain E about the asymptotically stable equilibrium $x = 0$, then $J(x_0)$ is bounded for all x_0 bounded and in the domain of stability.

Proof:

Given any x_0 in the domain of stability, there exists a bounded t_f such that $x(t_f)\ \varepsilon E$ and

$$J_f = \int_0^{t_f} (p + u'Ru)dt \tag{7}$$

is bounded. Within the domain E,

$$\bar{V} = -q(x) \leq -[p(x) + u'(x)R(x)u(x)] = J. \tag{8}$$

Since $x \to 0$ at $t \to \infty$ and $V(0) = J(0) = 0$,

$$V(x(t_f)) = \int_{t_f}^\infty q(x)dt \geq \int_{t_f}^\infty [p(x)+u'(x)R(x)u(x)] = J(x(t_f)). \tag{9}$$

But V is bounded in E, implying that J is also bounded in E. Hence $J(x_0)$ is bounded for all x_0 bounded and in the domain of stability.

3. Uniqueness of Solution

Applying the principle of optimality, we obtain the equation for the optimal cost function J(x) as

$$\min_u \left[p(x)+\tfrac{1}{2}u'R(x)u+J_x(x)(f(x)+B(x)u)\right] = 0, \quad J(0) = 0 \qquad (10)$$

where the subscript x denotes partial differentiation with respect to x. Assuming that p, R, f and B are sufficiently smooth and taking the partial derivative of equation (10) with respect to u, we obtain

$$R(x)u+B'(x)J_x'(x) = 0 \qquad (11)$$

which yields a nonlinear control law with a simple structure

$$u(x) = -R^{-1}(x)B'(x)J_x'(x). \qquad (12)$$

Substitution of u(x) into equation (10) results in the H-J equation

$$J_x(f-BR^{-1}B'J_x') = -(p+\tfrac{1}{2}J_xBR^{-1}B'J_x'), \quad J(0) = 0. \qquad (13)$$

We now show that the positive definite solution to equation (13) is unique and hence the problem (1), (2) has a unique optimal solution. A similar result exists in [5].

Theorem 1 (Uniqueness)

If Assumptions I-III are satisfied for all $x\varepsilon R^n$ and a continuously differentiable positive definite solution J(x) with J(0) = 0 to the H-J equation (13) exists for all $x\varepsilon R^n$, then u(x) of (12) is the unique optimal asymptotically stabilizing control.

Proof:

The asymptotically stabilizing property of the control u(x) of (12) follows immediately from equation (13) by considering J(x) as a Lyapunov function for system (1) controlled by u(x). Let $\bar{J}(x)$ with $\bar{J}(0) = 0$ be another continuously differentiable positive definite solution to the H-J equation such that

$$\bar{u}(x) = -R^{-1}(x)B'(x)\bar{J}_x'(x) \qquad (14)$$

is also an asymptotically stabilizing control. From equation (13), the difference $\delta J = J-\bar{J}$ satisfies

$$(\delta J)_x(f-BR^{-1}B'J_x') = -(p+\tfrac{1}{2}J_xBR^{-1}B'J_x')+(p+\tfrac{1}{2}\bar{J}_xBR^{-1}B'\bar{J}_x')$$

$$+\bar{J}_xBR^{-1}B'(\delta J)_x' \qquad (15)$$

$$= -\tfrac{1}{2}(\delta J)_xBR^{-1}B'(\delta J)_x', \qquad \delta J(0) = 0$$

which can be rewritten as a system of ordinary differential equations [12]

$$\dot{x} = f - BR^{-1}B'J'_x \tag{16}$$

$$\delta J = -\tfrac{1}{2}(\delta J)_x BR^{-1}B'(\delta J)'_x .$$

Since $u(x)$ of (12) is an asymptotically stabilizing control, the boundary conditions of (16) are $x=0$ at $t=\infty$ for all x at $t=0$. Since $\delta J=0$ at $t=\infty$, $\delta J(x)$ is uniquely determined by x. Since δJ is a nonincreasing function satisfying the condition $\delta J=0$ at $t=\infty$, $\delta J=J-\bar{J}\geq 0$. Similarly, we can show that

$$(-\delta J)_x(f-BR^{-1}B'\bar{J}'_x) = -\tfrac{1}{2}(\delta J)_x BR^{-1}B'(\delta J)'_x \tag{17}$$

implying $\bar{J} - J\geq 0$. Hence we conclude that $J=\bar{J}$ and $u(x)$ of (12) is indeed the unique optimal control.

A similar result can be obtained when there is only a limited domain of stability.

It is well known that the stabilizing solution to the Riccati equation yields the unique optimal solution to the linear regulator problem [13,14]. It is of practical interest that a similar result exists in this class of nonlinear regulators where the positive definite solution to the H-J equation yields the unique optimal control. In the next section, we show that the Newton-Lyapunov method for solving the Riccati equation can be extended to nonlinear regulators.

4. The Iterative Scheme

We now apply the iterative scheme in [10] to design suboptimal controls. The basic scheme is that given an admissible control, we evaluate J to obtain J_x and the new improving control is generated by (12). In addition to the improvement in J shown in [10], we also show that the improvement in J is quadratic in the control and the domain of stability remains unchanged.

The iteration procedure is as follows. At the (k+1)st step, given $u^{(k)}$, we compute

$$J^{(k)}(x_o) = \int_0^\infty (p+\tfrac{1}{2}u^{(k)'}Ru^{(k)})dt \tag{18}$$

for the system (1) control by $u^{(k)}$

$$\dot{x} = f+Bu^{(k)}, \quad x(0) = x_o . \tag{19}$$

The new feedback control is generated by

$$u^{(k+1)}(x) = -R^{-1}B'J_x^{(k)'}(x). \tag{20}$$

The initial control is the admissible control $u^{(0)}$ whose properties are given in Assumption IV.

The equations (18) and (19) can be rewritten as a linear first order partial differential equation

$$\dot{J}^{(k)}=J_x^{(k)}(f+Bu^{(k)})=-(p+\tfrac{1}{2}u^{(k)'}\ Ru^{(k)}) \tag{21}$$

which is now used to provide the following stabilizing and cost improving properties.

Theorem 2

If Assumptions I-IV are satisfied, then $u^{(k+1)}$ is an asymptotically stabilizing admissible control in R^n and the improvement in the cost function J it produces over $u^{(k)}$ is quadratic in $(u^{(k+1)}-u^{(k)})$.

Proof:

(Stabilizing property). Consider the positive definite function $J^{(k)}$ generated by the admissible control $u^{(k)}$. System (1) controlled by $u^{(k+1)}$ is

$$\dot{x} = f + Bu^{(k+1)}. \tag{22}$$

The time derivative of $J^{(k)}$ with respect to system (22) is

$$\begin{aligned}\dot{J}^{(k)} &= J_x^{(k)}(f + Bu^{(k+1)})\\ &= J_x^{(k)}(f + Bu^{(k)} + B(u^{(k+1)}-u^{(k)}))\\ &= -(p+\tfrac{1}{2}u^{(k)'}Ru^{(k)}) - u^{(k+1)'}R(u^{(k+1)}-u^{(k)}) \qquad (23)\\ &= -(p+\tfrac{1}{2}u^{(k+1)'}Ru^{(k+1)}+\tfrac{1}{2}(u^{(k+1)}-{}^{(k)})'R(u^{(k+1)}-u^{(k)}))\\ &<0.\end{aligned}$$

Thus $J^{(k)}$ is a Lyapunov function for system (23), guaranteeing that the equilibrium x=0 is asymptotically stable. Furthermore, it follows from the assumptions that the domain of stability is for all $x\varepsilon R^n$.

(Cost improving property). Substituting equation (21) for k+1 into equation (23), we obtain after rearranging

$$(J^{(k)}-J^{(k+1)})_x(f+Bu^{(k+1)})=-\tfrac{1}{2}(u^{(k+1)}-u^{(k)})'R(u^{(k+1)}-u^{(k)}). \tag{24}$$

We rewrite equation (24) as

$$d(J^{(k)}-J^{(k+1)})/dt=-\tfrac{1}{2}(u^{(k+1)}-u^{(k)})'R(u^{(k+1)}-u^{(k)}) \tag{25}$$

where x is governed by system (22). Thus $(J^{(k)}-J^{(k+1)})$ is a decreasing function with $J^{(k)}=J^{(k+1)}=0$ as $t\to\infty$, and the improvement in J is quadratic in $(u^{(k+1)}-u^{(k)})$.

The above results can be modified if our domain of stability D is not the whole space R^n. We replace Assumption IV by

V. There exists in D R^n a sufficiently smooth asympototically stabilizing control $u^{(0)}(x)$. It is admissible in the sense that the function J is bounded for any x(0) in the interior of D, and $J \to \infty$ as x approaches the boundary of D. In addition, the control law (12) possesses the property that $|u(x)| \to \infty$ as $|J_x| \to \infty$.

Corollary 1

If Assumptions I-III and V are satisfied, then $u^{(k+1)}$ is an asymptotically stabilizing admissible control for all x in the interior of D and the improvement in J over $u^{(k)}$ is quadratic in $(u^{(k+1)}-u^{(k)})$.

Proof:

The proof is the same as that of Theorem 2 except for the domain of stability argument. From Assumption V and equation (21), it follows that for all k, $J^{(k)} \to \infty$ and $|J_x^{(k)}| \to \infty$ as x approaches the boundary of D. Thus using $J^{(k)}$ as a Lyapunov function, (23) shows that the trajectories of system (22) remain in D as $J^{(k)}$ is decreasing along all the trajectories.

We emphasize here that given the control $u^{(0)}$, we can find an improving control possessing the same stabilizing properties. The number of iterations that can be carried out depends on the smoothness of the problem. We shall not investigate the convergence question here, but this algorithm is basically of Newton's type as shown by equation (25). Hence if the algorithm converges, the rate of convergence is quadratic and the solution satisfies the H-J equation (13). Furthermore, it follows from Theorem 1 that the solution it converges to is the unique optimal solution. In the linear quadratic case, we assume that $J^{(k)}$ is of the form $x'P^{(k)}x$ where $P^{(k)}$ is a positive definite matrix. Then the separation of x from equation (21) yields a matrix Lyapunov equation. This is the Newton-Lyapunov iteration scheme for finding the solution to the Riccati equation [11].

5. Examples

The numerical solution for each iteration can be obtained by using the method of characteristics. For the point $x(0) = x_o$, we find its trajectory governed by system (19), and thus compute not only $V^{(k)}(x_o)$, but also obtain at the same time $V^{(k)}$ for all the points on this trajectory. A large number of trajectories may be necessary to compute $V^{(k)}$ and generate a feedback control map at each iteration. For a large domain, the amount of data storage and computation may be prohibitive.

To reduce the storage requirement and the computation of many trajectories, suboptimal schemes such as fixing the structure of the optimal cost function J can be used. In the case of analytic functions, the fixed structure may be a polynomial.

$$J_r = \sum_{i,j}^{n} a_{2ij}x_ix_j + \sum_{i,j,k}^{n} a_{3ijk}x_ix_jx_k + \dots + \sum_{i_1,i_2,\dots,i_r}^{n} a_{ri_1i_2\dots i_r}x_{i_1}x_{i_2}\dots x_{i_r} \qquad (26)$$

where the subscript r denotes the degree of the polynomial. The quadratic terms in J_r yield the linear feedback terms in the control u which is defined accordingly as

$$u_r = -R^{-1}B'(J_r)'_x. \tag{27}$$

J is evaluated at sufficiently many points and the coefficients

$$a_{ri_1 i_2 \ldots i_r}$$

are computed using an interpolation scheme such as the least square method. In certain problems, some of the coefficients can be fixed. For example, in systems with dominant linear behavior, we fix a_{2ij} to be the solution of the Riccati equation corresponding to the linearized problem.

Consider the regulation of the nonlinear system

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1^3 + u \tag{28}$$

$$J = \tfrac{1}{2}\int_0^\infty (x_1^2 + x_2^2 + u^2)dt. \tag{29}$$

For this problem it is found in [3] that the optimal cost function J consists of only even power terms in its power series expansion. Hence we investigate the approximations of J by

$$J_2 = q_1 x_1^2 + q_2 x_1 x_2 + q_3 x_2^2,$$

$$J_4 = J_2 + \sum_{i=1}^{5} q_{i+3}\, x_1^{5-i} x_2^{i-1}, \tag{30}$$

$$J_6 = J_4 + \sum_{i=1}^{7} q_{i+8}\, x_1^{7-i} x_2^{i-1}$$

where q_i are the coefficients to be computed. This yields the corresponding controls

$$u_r = -(J_r)_{x_2}, \quad r = 2, 4, 6. \tag{31}$$

We shall also investigate the approximation of J by fixing

$$q_1 = q_3 = \sqrt{3}/2, \quad q_2 = 1, \tag{32}$$

which is the Riccati gain of the problem (28), (29) when the nonlinearity $-x_1^3$ in the $\dot{x}_2$ equation in (28) is neglected. In this case, the J_r's and controls are denoted by the overbar. The function J is evaluated inside the domain

$$D = \{x_1, x_2 : 0 \le |x_1| \le 2,\ 0 \le |x_2| \le 2\}. \tag{33}$$

Using the Riccati gain feedback as $u^{(0)}$, the solutions q_i for all cases converge in a few iterations. The resulting coefficients are shown in Table 1 and the values of J at various points using different controls are shown in Table 2. From these results, it is clear that including more terms in J_r produces better controls and forces the gains q_1, q_2 and q_3 to converge to $\sqrt{3}/2$, 1 and $\sqrt{3}/2$ respectively. Comparing the values of J using the controls with and without fixed gains, we conclude that the controls without the fixed gains are better when x(0) is far from the origin and J is large, while the controls with fixed gains are better when x(0) is close to the origin and J is small. This is due to the characteristics of the least square method. An inevitable drawback of this scheme is that due to the truncation imposed on J, we change the domain of stability at each iteration. For example, u_1 and $\bar{u}_1$ are globally stabilizing controls whereas the other controls are not.

6. Conclusions

We have considered a nonlinear regulator problem with the same control structure as the linear regulators. For this problem, we show that the stabilizing solution of the Hamilton-Jacobi equation is unique. Based on the simple structure of the control law, a Newton-Lyapunov iterative design is proposed which generates successively improving controls and still retains the same domain of stability. Numerical computation using suboptimal schemes is shown to be feasible. Thus two results applicable to linear regulator problems are now extended to this class of nonlinear regulator problems.

Table 1. Coefficients in the J_r, r=2,4,6, polynomial approximations of J, with and without fixed q_1, q_2 and q_3.

	J_2	$\bar{J}_2$	J_4	$\bar{J}_4$	J_6	$\bar{J}_6$
q_1	.9024	.8665	.6195	.8665	.7946	.8665
q_2	.3285	1.	.8157	1.	.9025	1.
q_3	.6356	.8665	.8574	.8665	.8395	.8665
q_4			.07097	.02249	−.04620	−.08784
q_5			−.1177	−.1431	−.2820	−.4012
q_6			−.06237	−.07412	−.1918	−.3306
q_7			−.03497	−.05374	−.07410	−.1365
q_8			−.01611	−.01964	−.01275	−.01693
q_9					.01616	.02164
q_{10}					.02777	.04745
q_{11}					.02750	.05372
q_{12}					.01552	.03275
q_{13}					.006537	.01590
q_{14}					.002073	.004774
q_{15}					.0004487	−.0002028

References

1. E.G. Al'brekht, "On the Optimal Stabilization of Nonlinear Systems," J. Appl. Math. Mech. (PMM), Vol. 25, 1961, pp. 1254-1266.
2. D.L. Lukes, "Optimal Regulation of Nonlinear Dynamical Systems," SIAM J. Control, Vol. 7, 1969, pp. 75-100.
3. Y. Nishikawa, N. Sannomiya and H. Itakura, "A Method of Suboptimal Design of Nonlinear Feedback Systems," Automatica, Vol. 7, 1971, pp. 703-712.
4. W.L. Garrard, "Suboptimal Feedback Control for Nonlinear Systems," Automatica, Vol. 8, 1972, pp. 219-221.
5. W.L. Garrard and J.M. Jordan, "Design of Nonlinear Automatic Flight Control Systems," Automatica, Vol. 13, 1977, pp. 497-505.
6. P.J. Moylan and B.D.O. Anderson, "Nonlinear Regulator Theory and an Inverse Optimal Control Problem," IEEE Trans. Automatic Control, Vol. AC-18, 1973, pp. 460-465.
7. A.A. Krasovsky, "A New Solution to the Problem of a Control System Analytical Design," Automatica, Vol. 7, 1971, pp. 45-50.
8. R.W. Brockett, "Lie Theory and Control Systems Defined on Spheres," SIAM J. Appl. Math., Vol. 25, 1973, pp. 213-225.
9. D.H. Jacobson, "On the Optimal Control of Systems of Quadratic and Bilinear Differential Equations," Proceedings of IFAC Sixth World Congress, Boston, U.S.A., 1975, Part 1, paper 51.4.
10. R.J. Leake and R.W. Liu, "Construction of Suboptimal Control Sequences," SIAM J. Control, Vol. 5, 1967, pp. 54-63.
11. D.L. Kleinman, "On an Iterative Technique for Riccati Equation Computations," IEEE Trans. Automatic Control, Vol. AC-13, 1968, pp. 114-115.
12. F. John, Partial Differential Equations, 2nd edition, Spring-Verlag, New York, 1975.
13. W.M. Wonham, "On a Matrix Riccati Equation of Stochastic Control," SIAM J. Control, Vol. 6, 1968, pp. 681-697.
14. V. Kucera, "A Contribution to Matrix Quadratic Equations," IEEE Trans. Automatic Control, Vol. AC-17, 1972, pp. 344-347.
15. P.J. Moylan, "Optimal Control of Nonlinear Systems," Ph.D. Thesis, University of Newcastle, Australia, 1972.

Table 2. Values of J for points inside the domain D using controls derived from J_r, r=2,4,6, with and without fixed q_1, q_2 and q_3.

(x_{10},x_{20})	u_2	$\bar{u}_2$	u_4	$\bar{u}_4$	u_6	$\bar{u}_6$
(2,2)	6.725	7.455	6.603	6.653	6.582	6.626
(2,0)	3.515	3.673	3.484	3.492	3.478	3.479
(-2,2)	5.111	4.965	4.950	4.950	4.948	4.948
(0,2)	3.292	3.264	3.195	3.208	3.183	3.180
(1,1)	2.000	2.107	1.978	2.006	1.952	1.954
(1,0)	.8302	.7777	.7648	.7667	.7604	.7607
(-1,1)	.8488	.8426	.8453	.8420	.8420	.8419
(0,1)	1.011	.8510	.8525	.8499	.8502	.8494

A NONLINEAR SERVO FOR LINEAR SYSTEMS

L. SHAW
Polytechnic Institute of New York
Brooklyn, New York 11201

ABSTRACT

This paper combines the nonlinear regulator ideas introduced in [1], for linear multivariable systems, together with the suggestion in [2] for the conversion of a regulator controller into a servo controller. The horizon-based servos described here seem to be superior to the nonlinear servos in [2] with regard to quality of performance (e.g., less overshoot for the same rise-time).

Although horizon-based controllers require extensive on-line computation, microprocessor advances are markedly reducing that disadvantage.

The use of state observers is also explored, since horizon-based controllers are inherently of state feedback form.

1. Introduction

We will begin with a formulation of the servo configuration in which a nonlinear state regulator can be used to speed the tracking response. Some results about nonlinear horizon-based regulators will then be summarized. Finally, a few simple numerical examples will demonstrate the qualities of these controllers.

Methods are well known for designing linear feedback servos to control the outputs of linear plants. Here we consider augmentations of such a linear controller by a nonlinear loop which speeds the response when tracking errors are large.

The nonlinear controllers considered here are those previously described for regulation of multivariable linear plants [1,3], and they can be viewed as providing state-dependent feedback gains.

Since the usual servo is limited to operation on the scalar tracking error (reference minus output), the use of a state-dependent nonlinear controller will generally require use of a state observer to obtain the error-state vector which is to be driven toward zero. Even in the rare cases where all states of the plant are directly observable, direct measurement of the error-state vector will be possible only when tracking step changes in the reference signal. When tracking higher order polynomial reference signals, derivatives of the reference signal (which are not directly measurable) are essential for determination of the error-state vector.

Figure 1 shows the usual linear servo configuration in which P(s) may include both the given plant and a tandem compensator. If the denominator of P(s) (of the form $s^n + a_1 s^{n-1} \ldots a_n$) has $a_n = a_{n-1} \ldots a_{n-m+1} = 0$, then P is a type-m system which can follow an (m-1) degree polynomial reference signal ($r(t) = \sum_{i=1}^{m} \alpha_i t^{i-1}$) with zero steady-state error ($\varepsilon(t) \rightarrow 0$). This kind of single-input, single-output system can be described by the state variable differential equations

$$\dot{\underline{x}} = A\,\underline{x} + \underline{b} \quad ; \quad y = \underline{c}'\underline{x} \tag{1}$$

where

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & 0 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 & 1 \\ 0 & 0 & -a_{n-m} & \cdot & \cdot & -a_1 \end{pmatrix} ; \quad \underline{b} = \begin{pmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{pmatrix} ; \quad \underline{c} = \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} ; \tag{2}$$

It is convenient to define a reference vector $\underline{r}(t)$ and an error-state vector $\underline{e}(t)$ as follows

$$\underline{r} = \begin{pmatrix} r \\ \dot{r} \\ \cdot \\ \cdot \\ r^{(n)} \end{pmatrix} ; \qquad \underline{e} = \underline{r} - \underline{x}. \tag{3}$$

It follows that the last (n–m) elements of $\underline{r}$ are zero and that $\underline{c}'\underline{e} = \varepsilon$, the scalar tracking error. Moreover, the special canonical form of A used in (2) implies

$$A\underline{r} = \dot{\underline{r}}. \tag{4}$$

Substitution of (3) and (4) into (1) yields the differential equations

$$\begin{aligned} \dot{\underline{e}} &= (A - \underline{b}\,\underline{c}')\underline{e} \\ \dot{\underline{e}} &= \bar{A}\underline{e} \end{aligned} \tag{5}$$

so that $\underline{e} \to \underline{0}$ if $\bar{A}$ is a stable matrix.

Figure 2 shows how the foregoing linear servo can be augmented by an additional nonlinear state regulator in order to effect a faster tracking response. This configuration postulates the use of a dynamic state observer which has knowledge of the first component of $\underline{e}$ and the nonlinear control input $u_{NL}(\hat{\underline{e}})$. The presence of the nonlinear loop changes the error-state vector equations to

$$\dot{\underline{e}} = \bar{A}\underline{e} + \underline{b}\,u_{NL}(\hat{\underline{e}}). \tag{6}$$

If $\hat{\underline{e}}$ were to equal $\underline{e}$ then equation (6) would be in the form of a state variable regulator problem for which horizon-based nonlinear controllers have been developed in order to achieve fast reduction of "large" initial $\underline{e}$-vectors.

2. Review of Regulator Results

References [1,3] describe nonlinear regulators for systems of the form

$$\dot{\underline{e}} = \bar{A}\,\underline{e} + B\underline{u} \tag{7}$$

with $(\bar{A},B)$ controllable. Those regulators are based on linear regulators of the receding horizon or virtual horizon type in which the horizon parameter acts like a scalar measure of the response time. The nonlinear versions make the horizons state-dependent, with "large" error states inducing nearer horizons and correspondingly faster reduction of the error state toward zero.

For example, the linear receding horizon controller takes the form

$$\underline{u}(t) = - R^{-1}B'W^{-1}(T)\underline{e}(t), \tag{8}$$

where R is a positive definite matrix, and W(T) is the positive definite solution of the linear matrix equation

$$\frac{dW}{dT} = BR^{-1}B' - \bar{A}W - W\bar{A}'; \; W(0) = 0. \tag{9}$$

At each instant of time t, the controller in (8) acts to drive $\underline{e}$ to $\underline{0}$ at a time (t+T). Choice of a smaller horizon parameter T results in a system which regulates faster, while requiring correspondingly larger control energy. (The importance of conserving control energy can be introduced by the choice of R.)

The nonlinear receding horizon controller makes the horizon distance a function of the state. It is possible to make this T($\underline{e}$) a decreasing function of the "size" of $\underline{e}$ in such a way that the resulting nonlinear system is asymptotically stable, and, as demonstrated by simulations, the system responds faster to larger initial $\underline{e}(0)$.

The state-dependent T($\underline{e}$) measures the size of $\underline{e}$ via a quadratic form which is natural for the system, namely

$$V(\underline{e},T) = \underline{e}'W^{-1}(T)\underline{e} \tag{10}$$

It can be shown that this positive definite function is decreasing in T (for fixed $\underline{e}$). Thus the desired T($\underline{e}$) relation is achieved when T is defined inplicitly by the constraint

$$V(\underline{e},T) = F(T) \tag{11}$$

for a suitable decreasing function F(T). (See Figure 3.)

It can be shown that if

$$\frac{dF}{dT} < \frac{dV}{dT} \tag{12}$$

whenever (11) is satisfied (i.e. at the intersection points in Figure 3) then the V in (10) is a Lyapunov function, and the closed loop nonlinear system is asymptotically stable. The references discuss methods for finding suitable F(T) functions which ensure (12). It is clear that if F(T) decreases monotonically from infinity to zero as T increases from T_m to T_M, then at least one solution to (11) exists for each $\underline{e}$, and the one for which T is largest will satisfy the stability condition (12).

3. Examples

Since analytical solution of (9) is difficult except for system of low dimension, numerical techniques will be most appropriate for finding W(T) and solving (11) for T($\underline{e}$) at each instant of time. While development of such programs is in progress, some very simple examples have been simulated using analytical solutions of (9).

Referring to (2) and (5), a "type 1" plant defined by

$$A = \begin{pmatrix} 0 & 1 \\ 0 & -5 \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 0 \\ 4 \end{pmatrix} \quad \underline{c} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad R = 1$$

results in

$$\overline{A} = \begin{pmatrix} 0 & 1 \\ -4 & -5 \end{pmatrix}$$

for which an explicit expression for W(T) is found in [1 and 3].

We first consider tracking step changes in the reference input when the plant states can be measured (no observer). Figures 4, 5 and 6 demonstrate step response properties of this nonlinear servo when

$$F(T) = d^2 \left(\frac{1-T}{T-0.2}\right)^2$$

for various step amplitudes and various values of d. Also shown are responses of the linear servo of Figure 1 and the linear version of Figure 2 when $T=T_{max}=1$. The ability of the nonlinear controller to respond relatively faster for larger step amplitudes is quite evident. Figure 6 shows how the maximum amplitude of the nonlinear control signal increases as the controller is made "more nonlinear."

Since the tracking of higher order polynomial reference signals will require the use of a state observer, it is of interest to examine this same step response example when an observer is used. With the observer output $\underline{\hat{e}}$ defined by

$$\dot{\underline{\hat{e}}} = F\underline{\hat{e}} + \underline{h}\ \varepsilon + \underline{b}u$$

we can choose $F=(A-\underline{h}\ \underline{c}')$ to have suitably negative eigenvalues to provide fast convergence of the observer error $\underline{\epsilon}=(\underline{e}-\underline{\hat{e}})$, which satisfies the differential equation:

$$\dot{\underline{\epsilon}} = F\,\underline{\epsilon}.$$

(The full-dimension observer has been used because it has been noted that in some control problems this causes less performance degradation than the reduced order observer [4].)

The effects of observer convergence rate on the behavior of the nonlinear controller were investigated by choosing

$$|sI - F| = (s + \beta)^2$$

for several values of β. Figure 7 shows the unusual results. The step response is more sluggish here, as compared to the perfect state observation case, for all β. As β increases, from 1 to 30 to 50, the response becomes slower and slower; but changing from $\beta=50$ to $\beta=150$ improves the speed of response!

Conclusion

It has been shown that a nonlinear, horizon-based regulation can be used to augment the tracking performance of a linear servo, more work is needed to examine the interrelationship of the observer convergence to a precise estimate of the error-state vector, and the servo convergence of the error-state vector to zero.

Acknowledgment

This work was supported in part by the Joint Services Electronics Programs, Contract F44620-74-C-0056. The simulation studies were carried out efficiently by Mr. H. Gambe.

References

1. Shaw, L., Sarlat, D. and Thomas, Y., "Synthesis of Nonlinear Controllers," IFAC, 7th World Congress, Helsinki, June 1978.

2. Sandor, J. and Williamson, D., Nonlinear Feedback to Improve the Transient Response of a Linear Servo. IEEE Trans. on Automatic Control, 22, Oct. 1977, 863-864.

3. Shaw, L., "Nonlinear Control of Linear Multivariable Systems via State Dependent Feedback Gains," 1978 Conference on Decision and Control, San Diego, Jan. 1979.

4. Bongiorno Jr., J.J., and Youla, D.C., "On Observers in Multi-Variable Control Systems, Int. J. Control 8, 3, 221-243.

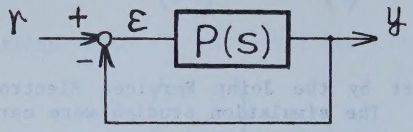


Fig-1. FEEDBACK SERVO CONFIGURATION

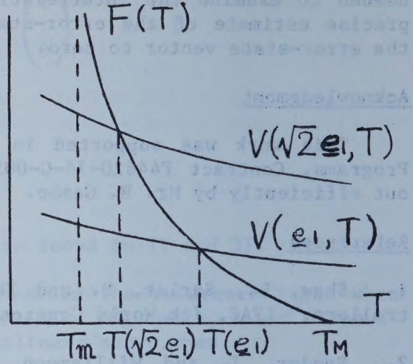

F(T)
V(√2 e₁, T)
V(e₁, T)
T
Tm
T(√2 e₁)
T(e₁)
TM

Fig-3. IMPLICIT DEFINITION of $T(\underset{\sim}{e})$

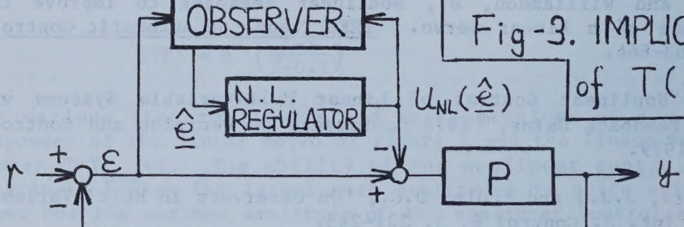

OBSERVER
N.L. REGULATOR
ê
U_NL(ê)
r
+
−
ε
−
+
P
y

Fig-2. SERVO with NONLINEAR AUGMENTATION LOOP

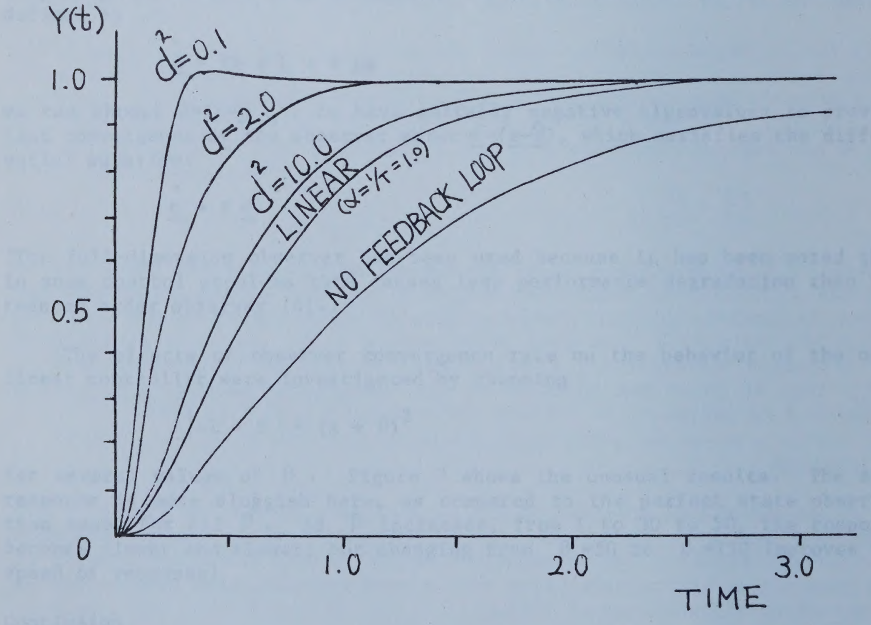


Fig-4. DIFFERENT NONLINEAR STEP RESPONSES

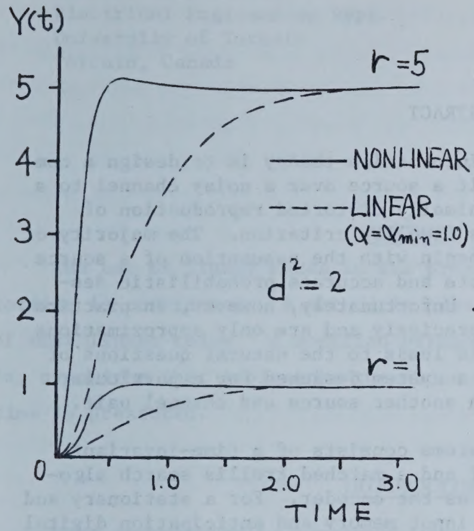


Fig-5. NONLINEAR STEP RESPONSE

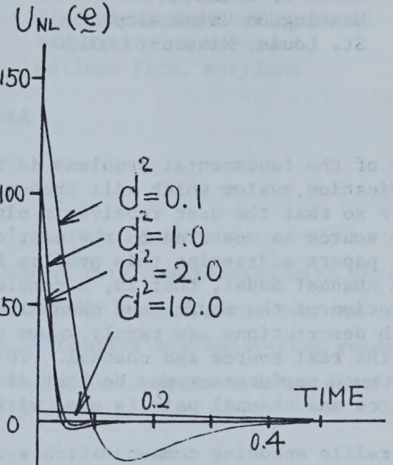


Fig-6. CONTROL AMPLITUDE for DIFFERENT NONLINEARITIES

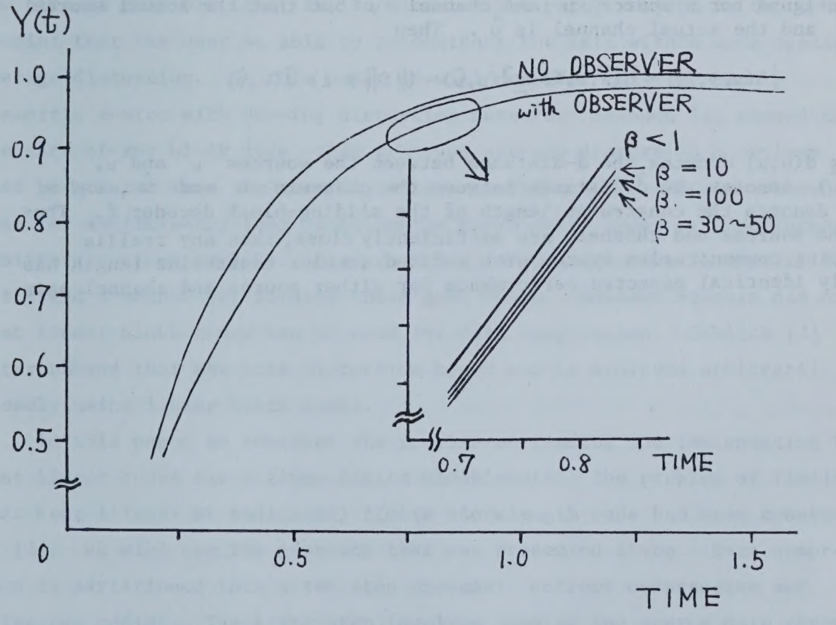


Fig-7. NONLINEAR STEP RESPONSE with OBSERVER

MISMATCH THEOREMS FOR TRELLIS ENCODING COMMUNICATION SYSTEMS

JAMES G. DUNHAM
Washington University
St. Louis, Missouri 63130

ABSTRACT

One of the fundamental problems in information theory is to design a communication system which will transmit a source over a noisy channel to a user so that the user receives a minimally distorted reproduction of the source as measured by the user's fidelity criterion. The majority of the papers addressing this problem begin with the assumption of a source and channel model, that is, a complete and accurate probabilistic description of the source and channel. Unfortunately, however, in practice such descriptions are rarely known precisely and are only approximations to the real source and channel. This leads to the natural questions of how much performance may be lost if a system designed for a particular source and channel pair is used with another source and channel pair.

A trellis encoding communication systems consists of a time-invariant (stationary) sliding-block decoder f and a matched trellis search algorithm such as the Viterbi algorithm as the encoder. For a stationary and ergodic discrete source μ, a finite input memory and anticipation digital channel ν and the Hamming distance per-letter fidelity criterion d, let $\Delta(\mu,\nu,f)$ denote the expected performance of the trellis encoding communication system formed from the source μ, channel ν, decoder f and optimal trellis encoder. Suppose that the trellis encoding communication system is designed for a source μ and channel ν but that the actual source is $\hat{\mu}$ and the actual channel is $\hat{\nu}$. Then

$$|\Delta(\mu,\nu,f) - \Delta(\hat{\mu},\hat{\nu},f)| \leq \bar{d}(\mu,\hat{\mu}) + (\|f\| + 1)\, \bar{d}(\nu,\hat{\nu})$$

where $\bar{d}(\mu,\hat{\mu})$ denotes the $\bar{d}$-distance between the sources μ and $\hat{\mu}$, $\bar{d}(\nu,\hat{\nu})$ denotes the $\bar{d}$-distance between the channels ν and $\hat{\nu}$, and $\|f\|$ denotes the constraint length of the sliding-block decoder f. Thus if the sources and channels are sufficiently close, then any trellis encoding communication system with a fixed decoder constraint length has nearly identical expected performance for either source and channel pair.

ENTROPY-CODED GROUP CODES FOR DATA COMPRESSION

ALBERTO LEON-GARCIA
Electrical Engineering Dept.
University of Toronto
Toronto, Canada

LEE D. DAVISSION
Electrical Engineering Dept.
University of Maryland
College Park, Maryland

ABSTRACT

The use of linear codes in the problem of data compression with distortion is considered. A method is presented for reducing the average rate of most linear codes. A modified version of Goblick's step-by-step encoding algorithm that implements the above method and reduces average search time is presented.

INTRODUCTION

Data compression is concerned with the efficient representation of source data prior to storage or transmission. In order to reduce storage and transmission costs, it is desired to represent the source data with as few bits as possible (that is, to minimize the rate) subject to the constraint that the user be able to reconstruct the data within some specified average distortion. In this paper we will consider only the binary symmetric source with Hamming distortion measure. Shannon [1] showed that the rate of any block code that achieves average distortion D or less must be greater than the rate distortion function R(D). He also showed that for sufficiently long blocklengths there exist codes with performance arbitrarily close to R(D). His proof was not constructive in that it did not yield a method for finding these good codes. However Shannon did note that linear block codes can be used for data compression. Goblick [2] later showed that the rate distortion bound can be achieved arbitrarily closely using linear block codes.

In this paper we consider the problem of finding and implementing the best linear codes for a given finite blocklength. The problem of finding that best (linear or nonlinear) finite blocklength code has been considered in [3]. We will use the approach that was presented there. Data compression is partitioned into a two step process: entropy compression and noiseless coding. The first step involves passing the source data through an "entropy compressor" that, by introducing distortion in a controlled manner, yields a reproduction sequence of lower entropy. The second step involves translating this lower entropy into lower average rate by encoding the reproduction sequence using a variable length code.

For block codes the problem involves finding a codeword set C and a mapping from the source n-tuples into C in such a way that the entropy of the codeword set is minimized subject to the given average distortion constraint. A good code must be efficient in trading off distortion for entropy. This involves balancing the following two conflicting requirements: In order to minimize average distortion, the source outputs must be mapped into the nearest codeword; In order to minimize entropy, the size of the codeword set must be minimized and the probability mass function over the codewords "tilted" as much as possible. The problem can be solved using a simplex-type algorithm [3]. The resulting codes however do not posses a structure that faciliates implementation.

LINEAR CODES

All applications of linear codes to the data compression problem with distortion involves using linear codes "backwards". The source emits a block $\underline{x}$ of length n and the source encoder (the channel decoder of the given code) treats $\underline{x}$ as if it were the output of a noisy channel and attempts to find the nearest codeword. The codeword can be specified using $k < n$ bits, so the data is compressed by a factor of k/n at a cost of introducing some distortion.

We now summarize some facts about linear codes [4]. An (n,k) linear code C is a k-dimensional subspace of the vector space of binary n-tuples over GF(2). The code has 2^k codewords and is specified by giving a basis for the subspace. The generating matrix G of a code consists of the matrix with these basis vectors as rows. Any codeword $\underline{c}$ can then be specified by a k-dimensional vector $\underline{m}$ as follows: $\underline{c} = \underline{m}\, G^t$. The code C can also be specified by its parity check matrix H which is defined to be the generating matrix of the (n-k)-dimensional subspace orthogonal to C. Thus $\underline{c}$ is a codeword iff $H\underline{c}^t = \underline{0}$.

The set of 2^n binary ntuples can be partitioned into 2^{n-k} disjoint cosets of size 2^k. A coset consists of the translation of the code C by some vector $\underline{e}$: $\underline{e} + C = \{\underline{e} + \underline{c} : \underline{c} \in C\}$. Two vectors $\underline{e}_1$ and $\underline{e}_2$ are in the same coset iff $\underline{e}_1 + \underline{e}_2 \in C$. The syndrome $\underline{s}$ of a vector $\underline{x}$ is defined by $\underline{s} = H\underline{x}^t$. All vectors in the same coset have the same syndrome and there is a one-to-one correspondence between syndromes and cosets. In particular, the code C is the coset corresponding to the syndrome $\underline{0}$.

The channel decoder's task is to find the codeword closest to the received vector $\underline{x}$. This is equivalent to finding the minimum weight error pattern $\underline{e}$ such that $\underline{x} = \underline{c} + \underline{e}$, where $\underline{c} \varepsilon C$. All error patterns must belong to the coset containing $\underline{x}$ since $H^t\underline{x} = H^t(\underline{c} + \underline{e}) = H^t\underline{e}$. The channel decoder must therefore find a minimum weight element in the coset containing $\underline{x}$. Note that the minimum weight element need not be unique. In channel decoding a minimum weight element is chosen to be the coset leader. The decoder has a table listing the syndromes and the corresponding coset leaders. Decoding proceeds as follows: Compute the syndrome $\underline{s} = H\underline{x}^t$; Look up the corresponding error pattern (coset leader) $\underline{e}(\underline{s})$; The codeword estimate is $\underline{c} = \underline{x} + \underline{e}(\underline{s})$. The standard array table contains all of the above information: The cosets form the rows of the array with the coset leaders listed in the first column; The code is the first row and the coset elements are arranged so that they are listed in the column of the codeword into which they are decoded. Figure 1 contains an example.

0000	1001	0110	1111
1000	0001	1110	0111
0100	1101	0010	1011
0011	1010	0101	1100

Figure 1

If the above scheme is used for source encoding the performance will be as follows. The source is symmetric so all n-typles have probability 2^{-n}. Since each element of a coset is mapped into a different codeword, the codewords will be equiprobable with probability 2^{-k}. The codeword set then has an entropy per source bit equal to $n^{-1}\log(1/2^{-k}) = k/n$. The cosets are also equiprobable with probability $2^{-(n-k)}$. Letting $\omega(s)$ be the weight of the minimum weight element in the coset corresponding to $\underline{s}$, the average distortion per source symbol is

$$D = 2^{-(n-k)} n^{-1} \sum_{\underline{s}} \omega(\underline{s}).$$

Thus the example in figure 1 has rate 1/2 and average distortion 1/4.

The above scheme is inefficient in trading distortion against entropy. Our first result is a method for improving the entropy performance of any linear code whose standard array contains cosets with non-unique coset leaders. The basic idea is simple: When the coset leader is not unique the source block can be mapped into several codewords; The encoder can use these choices to tilt the probability mass function of the codeword set. The method involves modifying the standard array construction as follows:

(1) Construct a reduced standard array containing as rows only the cosets with unique coset leaders.

(2) If a coset has ℓ minimum weight elements list ℓ rows using each of the minimum weight patterns as coset leaders, deleting all minimum weight patterns from all columns except the first.

(3) Select the column with the most elements below it. All blocks in this column will be mapped into the codeword at the top of the column. Cross out the n-tuples listed in this column from all other columns in the array.

(4) Repeat step (3) excluding all columns that have already been selected.

This procedure will result in nonuniformly distributed codewords whenever any of the coset leaders is not unique. Since the usual standard array yields uniformly distributed codewords and since the uniform distribution has maximum entropy, the codeword set obtained by the above method must have lower entropy.

0000	1001	0110	1111
1000		~~1110~~	0111
0001		~~0111~~	1110
0100	~~1101~~		1011
0010	~~1011~~		1101
0011			
1010			
0101			
1100			

Figure 2

Figure 2 illustrates the procedure for the code introduced in Figure 1. An exhaustive search was carried out for n = 3,4 and a partial search was carried out for n = 5. Table and Figure 3 summarize the results.

DECODING ALGORITHMS

In this section we present an algorithm for implementing the modified standard array construction presented above. Assume that the decoder has a table listing syndromes and the weights of the minimum weight elements in the corresponding cosets. Order the codewords in order of decreasing probability: $\{\underline{c}_1, \underline{c}_2, \ldots, \underline{c}_{2^k}\}$. The algorithm works as follows:

Compute $\underline{s} = H\underline{x}^t$

If $\underline{s} = \underline{0}$, $\hat{\underline{x}} = \underline{x}$.

If $\underline{s} \neq \underline{0}$, let $D_o = \omega(\underline{s})$.

```
    $i = 0$
  → $i = i + 1$
    If weight $(\underline{x} + \underline{c}_i) = D_o$ go to *
* $\hat{\underline{x}} = \underline{c}_i$
```

The algorithm searches for a coset element $\underline{x} + \underline{c}_i$ of minimum weight. Because the ordering of the codewords corresponds to the manner in which the columns were selected in the modified standard array construction, $\underline{x}$ will be mapped into the same $\underline{c}_i$ as in the modified standard array method.

The average search time $\bar{\ell}$ is defined to be the average number of iterations necessary to find $\hat{\underline{x}}$ when $\underline{s} \neq \underline{0}$. The average search time can be readily computed from the modified standard array. Note that the average search time decreases as the "tilt" in the codeword probability mass function is increased. Table 1 lists the average search times for the codes.

The above algorithm is related to the Slepian-Moore-Prange [5] step-by-step channel decoding algorithm and is a modification of Goblick's [2] step-by-step source encoding algorithm. The former implements a single-coset-leader standard array table. The latter (see [6, pg. 207]) involves a nonexhaustive search through the rows of the generating matrix. It does not use the coset weight information. Goblick computed the performance of the scheme for randomly generated codes of various blocklengths. Figure 3 shows the performance for blocks of length 20. It can be readily seen that the codes presented here of blocklengths n = 3,4,5 already have comparable performance.

As k increases the exhaustive search through the codeword set becomes unfeasible and a suboptimal version becomes desirable. One approach is to terminate the search after a certain number of codewords have been tested. If a codeword has not been found up to that point then the $\underline{x}$ is mapped into the nearest codeword tested. This in essence involves a further modification of the standard array. The resulting code will have lower entropy and average search time and higher distortion. This method was applied to the n=5, $\bar{\ell}=4.29$ code of Table 1. The code with the asterisk Table 1 is the code that results when the search is terminated after 4 steps.

Another approach would be to terminate the search as soon as a codeword with distortion less than some threshold D_T is found. This approach would eliminate the need for the table listing the syndromes and coset weights. Yet another approach would be to combine the above two modifications.

DISCUSSION

The outstanding problem in using linear codes to achieve data compression has been to find "complete" encoding algorithms that always find the nearest codeword. The only method that works for all linear codes, the standard array method and its step-by-step implementation, becomes unfeasible as the blocklength increases. Research in the area has concentrated on finding complete encoding algorithms for specific classes of codes. Complete algorithms are available for Hamming codes, double-error-correcting BCH codes, and "most" triple-error correcting BCH codes [7]. Figure 3, shows the performance of these codes. It remains to be seen whether the method presented here can be used to find codes of comparable performance and ease of implementation.

REFERENCES

[1] C.E. Shannon, "Coding Theorems for a Discrete Source with a Fidelity Criterion", 1959 IRE National Conventional Record, part 4, pp. 142-163.

[2] T.J. Goblick, "Coding for a Discrete Information Source with a Distortion Measure", Ph.D. dissertation, Department of Electrical Engineering, M.I.T., Cambridge, Mass.

[3] A. Leon-Garcia, "A Constructive Approach to Rate Distortion Theory", Proceedings of the 1978 International Conference on Communications, 34.5.1 - 34.5.3.

[4] F.J. MacWilliams and N.J.A. Sloane, The Theory of Error Correcting Codes, North Holland Publishing Co., New York, 1977.

[5] W.W. Peterson and E.J. Weldon, Error Correcting Codes, The MIT Press, Cambridge, Mass. 1972.

[6] Toby Berger, Rate Distortion Theory, Prentice Hall, Englewood Cliffs, N.J. 1971.

[7] Jose Van der Horst and Toby Berger, "Complete Decoding of Triple Error Correcting Binary BCH Codes", IEEE Trans. on Info. Th., vol. IT-22, No.2, pp. 138-146.

n	R=k/n	H/n	D	$\overline{\ell}$
3	.667	.583	.167	1.25
3	.333	.333	.250	1.50
4	.750	.637	.125	1.50
4	.500	.373	.250	1.33
4	.250	.224	.313	1.28
5	.800	.705	.100	2.12
5	.600	.587	.150	4.29
* 5	.600	.484	.200	2.83
5	.600	.487	.200	1.91
5	.400	.391	.225	2.21
5	.400	.355	.250	1.82
5	.200	.200	.313	1.50

Table 1

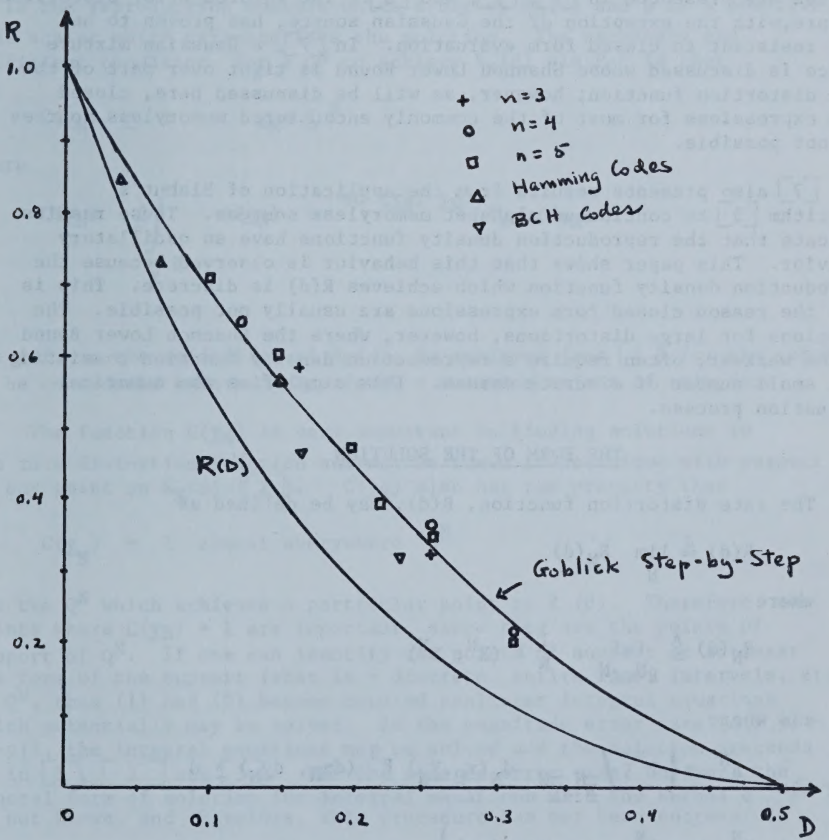


Figure 3

RATE DISTORTION FUNCTIONS FOR SQUARED ERROR DISTORTION MEASURES

STEPHEN L. FIX
General Electric Company
Utica, New York

ABSTRACT

A summary of recent results on the rate distortion function for a squared error distortion measure is presented. The fact that for a majority of the commonly encountered stationary memoryless sources that the reproduction probability density function which achieves R(d) is discrete is discussed.

INTRODUCTION

Considerable effort has been expended over the past twenty years toward finding ways of evaluating and bounding the rate distortion function. Berger [4] covers the effort through 1971. Recently, progress has been made for the magnitude error distortion measure and closed form expressions for the rate distortion function for particular sources are presented in [1], [2], and [3]. The squared error distortion measure, with the exception of the Gaussian source, has proven to be more resistant to closed form evaluation. In [7], a Gaussian mixture source is discussed whose Shannon Lower Bound is tight over part of the rate distortion function; however, as will be discussed here, closed form expressions for most of the commonly encountered memoryless sources are not possible.

[7] also presents results from the application of Blahut's algorithm [5] to continuous alphabet memoryless sources. These results indicate that the reproduction density functions have an oscillatory behavior. This paper shows that this behavior is observed because the reproduction density function which achieves R(d) is discrete. This is also the reason closed form expressions are usually not possible. The solutions for large distortions, however, where the Shannon Lower Bound is the weakest, often require a reproduction density function consisting of a small number of discrete masses. This simplifies the numerical evaluation process.

THE FORM OF THE SOLUTION

The rate distortion function, R(d), may be defined as

$$R(d) \triangleq \lim_{N} \; R_N(d)$$

where

$$R_N(d) \triangleq \inf_{P^N \varepsilon A^N} \; \frac{1}{N} I\,(X^N;\, Y^N)$$

and where

$$A^N = \left\{ P^N : \int_{\underline{X}^N x \underline{Y}^N} d_N(\underline{x}_N, \underline{y}_N)\; P^N\,(d\underline{x}_N,\, d\underline{y}_N) \leq d \right.$$

$$\left. \text{and } \; P^N \; \text{ has } P^N_x \; \text{ as a marginal} \right\}$$

P^N is from the set of all joint probability measures on the product space $\underline{X}^N \times \underline{Y}^N$ with $P_x{}^N$ as the probability measure on $\underline{X}^N$. $P_x{}^N$ is the source probability measure, $I(\cdot,\cdot)$ is mutual information, and $d_N(\cdot,\cdot)$ is the distortion criteria. The Y^N marginal, Q^N, is known as the reproduction measure. Here, $d_N(\underline{x}_N, \underline{y}_N)$ will be given by

$$d_N(\underline{x}_N, \underline{y}_N) = \frac{1}{N} \sum_{i=1}^{N} d(x_i, y_i)$$

where

$$d(x\ y) = (x - y)^2$$

The form which P^N must have if it achieves $R_N(d)$ is well known (see [4] or [8]) to be $P_Q{}^N(d\underline{x}_N, d\underline{y}_N) = a(\underline{x})\ e^{-sd_N(\underline{x}_N, \underline{y}_N)}$ $P_X{}^N \times Q^N(d\underline{x}_N, d\underline{y}_N)$

where

$$a(\underline{x}_N) \triangleq \int_{\underline{Y}^N} e^{-sd_N(\underline{x}_N, \underline{y}_N)}\ Q^N(d\underline{y}_N) \qquad (1)$$

Q^N is the reproduction measure which achieves $R_N(d)$ and s is a positive real scalar which parameterizes the solution. The necessary and sufficient condition for a Q^N to achieve $R_N(d)$ via $P_Q{}^N$ is that

$$C(\underline{y}_N) \leq 1 \qquad \underline{y}_N \forall \underline{Y}^N$$

where

$$C(\underline{y}_N) \triangleq \int_{\underline{X}^N} a(\underline{x}_N)\ e^{-sd(\underline{x}_N, \underline{y}_N)}\ P_x(d\underline{x}_N) \qquad (2)$$

The inf, in the definition of $R_N(d)$, is achieved (see [2]) for the cases to be considered here and therefore, a solution exists of this form.

The function $C(\underline{y}_N)$ is very important in finding solutions to the rate distortion function and may be shown to be unique with respect to any point on $R_N(d)$ ([2]). $C(\underline{y}_N)$ also has the property that

$$C(\underline{y}_N) = 1 \text{ almost everywhere } Q^N$$

for the Q^N which achieves a particular point on $R\ (d)$. Therefore, the points where $C(\underline{y}_N) = 1$ are important, since they are the points of support of Q^N. If one can identify the points of support or at least the form of the support (that is - discrete, entire line, intervals, etc) of Q^N, then (1) and (2) become coupled nonlinear integral equations which potentially may be solved. In the magnitude error case ($d(x,y) = |x-y|$), the integral equations may be solved and the solution proceeds as in [1], [2] and [3]. For the squared error case, however, the general form of solution for integral equations with the kernel $e^{-s(x-y)^2}$ is not known, and therefore, this procedure has not been successful.

For the squared error distortion measure for the case of N = 1 (memoryless source), one has

$$a(x) = \left[\int_{\underline{Y}} e^{-s(x-y)^2} Q(dy)\right]^{-1}$$

and

$$C(y) = \int_{\underline{X}} a(x) \; e^{-s(x-y)^2} \; P_X(dx)$$

Although the general form of solution of integral equations of this type is not known, the kernel does have properties which dictate the form of solution in this case. To be specific, $e^{-s(x-y)^2}$ is a real entire function of y and it can be shown [2] that the C(y) associated with any point on R(d) is also a real entire function. Therefore, the real zeros of the real entire function C(y)-1 are the points of support of the Q which yields R(d) at the point d. Liouville's theorem thus demands that the support of Q either be the entire real line or at most, a countable number of isolated points, but not both. Further, source densities which have support themselves only over a portion of the real line, such as the uniform or the Rayleigh source densities, must have Q's which have support only over a portion of the real line. The support of Q for such sources therefore, must be discrete. This situation always occurs when a source density is truncated for use in a numerical evaluation of R(d) via Blahut's algorithm [5]. This explains the "oscillations" in the density of Q observed in [7] when Blahut's algorithm was applied to a truncated Laplacian source.

EXAMPLES

The discrete support of Q can influence the rate distortion function evaluation procedure. Procedures for taking advantage of the discrete nature of the density of Q are described in [2] and [6] and take advantage of the derivative of C(y) to update the positions of the support points.

To understand how Q varies as a function of d, it is interesting to start at the point $d_{max}(R(d_{max}) = 0)$. The Q which achieves the point at d_{max} is a single point mass at the mean of the source. As d decreases from d_{max}, more point masses are needed. The masses develop as d decreases by two processes:

- They may occur by an existing mass splitting.
- They may grow.

Figure 1 shows for the zero mean, unit variance, uniform density, memoryless source how the mass positions evolve with d. As can be seen, two masses form by splitting from the single mass at d_{max}. The two mass solution continues as d decreases, with the two masses propagating outward from the mean. This continues until a third mass starts to grow at the mean. This three point solution continues, as shown in the figure, until two more masses start to grow, causing a five point solution. The situation is better visualized by considering C(y). For the example being considered, sketches of C(y) at various stages are shown in Figure 2. It is important to note that new points are needed as d decreases to keep C(y) under one. That is, as the two point solution spreads farther apart to satisfy the needs of the outer parts of the density function, the inner region requires restrengthening. C(y) will begin to form a peak which will eventually need a point mass to control it. This evolution is demonstrated in Figure 2 for the uniform source density.

This example illustrates the process by which the discrete masses which make up Q evolve. It is noted in the uniform source density example that 68% of the distortion axis is generated by a Q with only two (both weight .5) points of support (See [2]). This situation is not always the case however. The truncated one-sided exponential source, for example, with source density given by

$$p_x(x) = \frac{e^{-x}}{\left[1-e^{-10}\right]} \qquad x\varepsilon \left[0,10\right]$$

has a two point solution region of approximately 14% of d. The truncated Laplacian source given by

$$p_x(x) = \frac{e^{-\sqrt{2}\,|x|}}{\sqrt{2}\left[1-e^{-\sqrt{2}\,10}\right]} \qquad x\varepsilon \left[-10, 10\right]$$

has no two point solution region, but instead, goes from the situation at d_{max} to a three point solution by growing two new masses.

Figure 3 shows how the function C(y) develops as d decreases from d_{max} through the two point solution region for the one sided exponential source. As can be seen, the second point of support grows and propagates inward as d continues to decrease. This is in contrast to the way the support points evolve in the uniform case. In either case, however, the development of the points of support can be predicted by inspection of C(y) as has been demonstrated here.

To get an idea of the growth of the points of support of Q with distortion, Jensen's theorem on the number of zeros of an entire function within a disk of a certain radius may be used, [2]. Application of this indicates that the number of points of support in a finite radius disk can grow no faster than linearly with s. Since, for a truncated source, the extent of the support of Q cannot exceed that of the source truncation, this bound may be used to bound the total number. To roughly relate s to d, it should be remembered ([2] or [4]) that

$$d < \frac{1}{2s}$$

CONCLUSIONS

The Gaussian source is an example of a source whose reproduction measure has the entire real line of support. Unfortunately, most of the usual memoryless sources encountered have the discrete alternative. These include sources with one sided exponential, Rayleigh, Ricean, and uniform density functions. The form that the solution must take for these sources has been illustrated here.

When a source with memory is considered (N>1), C(y) continues to place requirements on the form of the solution. The theorems on entire functions of N-dimensions, however, do not restrict the support of Q^N to isolated discrete points. They do, however, restrict the form of the solution. Whether these requirements will simplify the solution of $R_N(d)$ for sources with memory is unclear at the present.

REFERENCES

1. Tan, H. H. and Yao K. (1975), "Evaluation of Rate Distortion Functions for a Class of Independent Identically Distributed Sources under an Absolute Magnitude Criterion". Trans. IEEE, IT-21, No. 1, January 1975, p. 59-64.
2. Fix, S.L., (1977), "Rate Distortion Functions for Continuous Alphabet Memoryless Sources", PhD Dissertation, University of Michigan, September 1977.
3. Yao, K. and Tan H. H. (1978), "Absolute Error Rate-Distortion Functions for Sources with Constrained Magnitudes" Trans. IEEE, IT-24, No. 4, July 1978, p. 499-503.
4. Berger, T., (1971), Rate Distortion Theory: A Mathematical Basis for Data Compression, Prentice-Hall.
5. Blahut, R. E. (1972) "An Hypothesis Testing Approach to Information Theory", PhD Dissertation, Cornell University, August 1972.
6. Benjamin T. W. (1973), "Rate Distortion Functions for Discrete Sources with Continuous Reproductions", Master's Thesis, Cornell University
7. Wilson, S. G. and Lytle, D. W. "Trellis Encoding of Continuous-Amplitude Memoryless Sources", Trans. IEEE, IT-23, No. 3, May 1977, p. 404-409.
8. Csiszar, I. (1974), "On an Extremum Problem of Information Theory", Studia Scientiarium Mathematicarum Hungarica, 9, 1974, pg. 57-71.

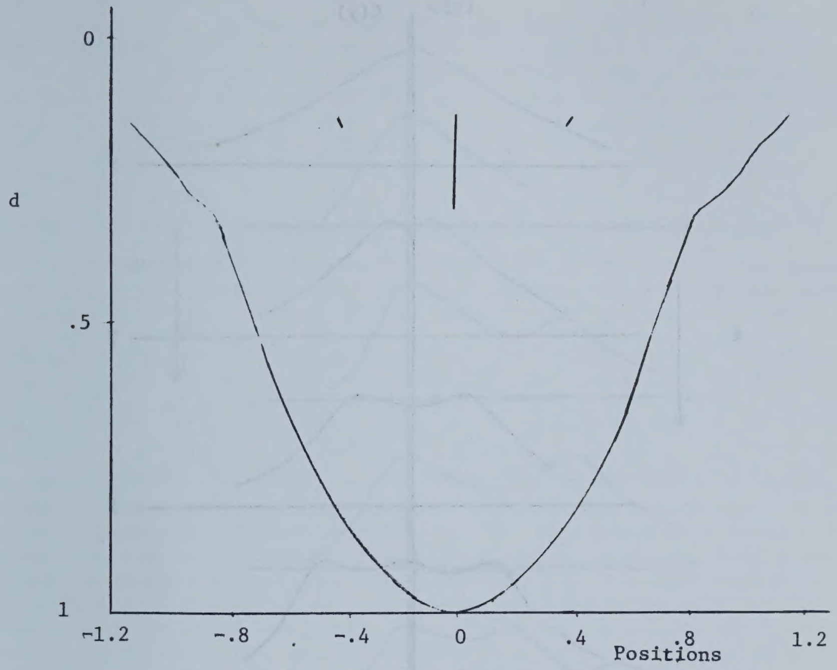


Figure 1. The Positions of the Support of Q Versus d for the Unit Variance Uniform Source.

Rate Distortion Function for the Unit Variance Uniform Source

s	d	R	s	d	R
.6	.7489	.1369	1.6	.2887	.5428
.7	.6024	.2315	1.7	.2687	.5758
.8	.5106	.3	1.8	.2552	.5992
.9	.4499	.3515	1.9	.2375	.6318
1.	.4077	.3915	2.	.2249	.6563
1.1	.3777	.4228	2.1	.2141	.6786
1.2	.3555	.4483	2.2	.2046	.6989
1.3	.3388	.4691	2.3	.1959	.7185
1.4	.3255	.4871	2.4	.1890	.7346
1.5	.3118	.5070	2.6	.1770	.7648
			3.4	.138	.8811

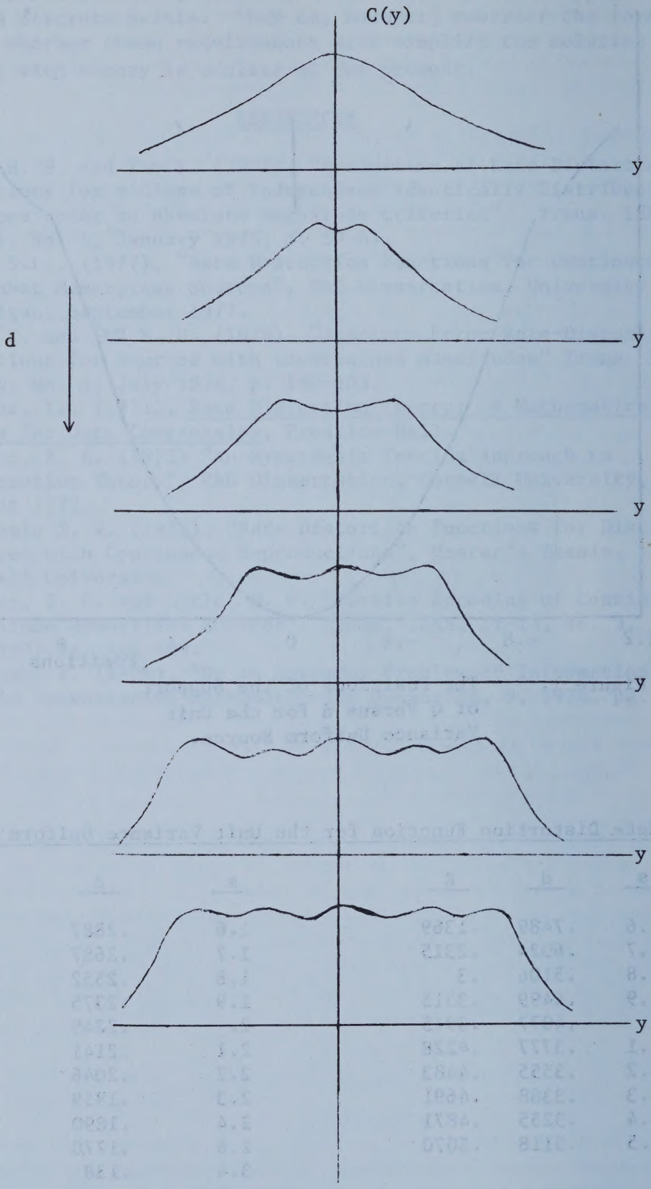


Figure 2. C(y) for the Uniform Source As d Decreases.

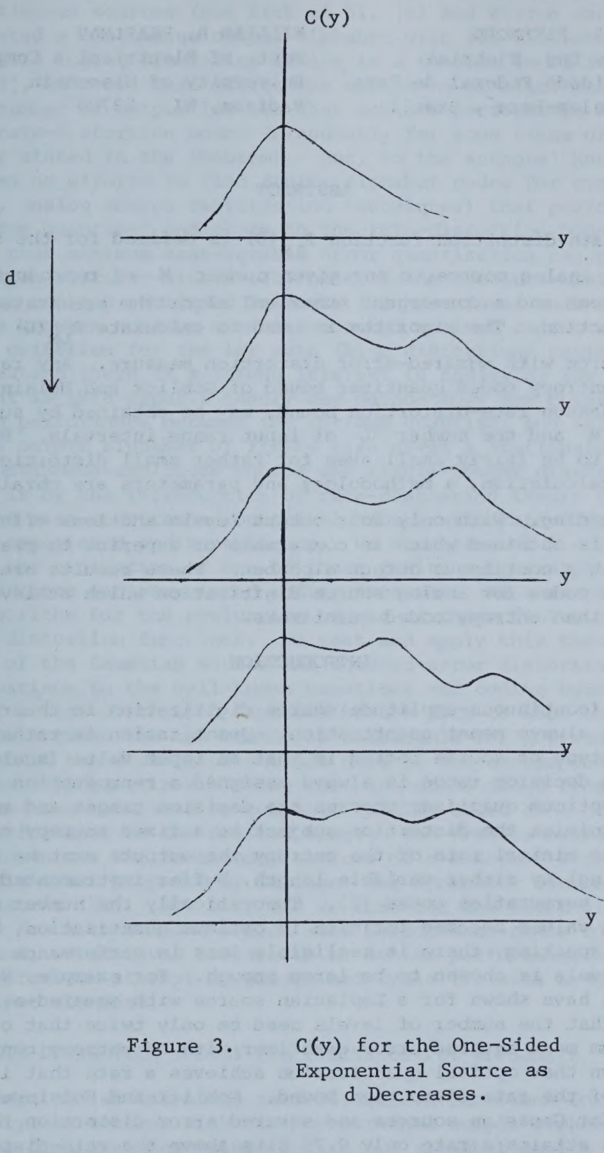


Figure 3. C(y) for the One-Sided Exponential Source as d Decreases.

A NEW RATE-DISTORTION FUNCTION AND NEW CODING RESULTS FOR ANALOG SOURCE DIGITIZATION

WEILER A. FINAMORE
Dept. de Eng. Eletrica
Universidade Federal de Pará
66000-Belem-Pará, Brazil

WILLIAM A. PEARLMAN
Dept. of Electrical & Computer Engr.
University of Wisconsin
Madison, WI 53706

ABSTRACT

A new rate-distortion function $R_{LM}(D)$ is defined for the digitization of an i.i.d. analog source to any given number M of reproduction levels. Coding theorems and a convergent numerical algorithm are stated and proved for this function. The algorithm is used to calculate $R_{LM}(D)$ for the Gaussian source with squared-error distortion measure. Any rates between $R^*(D)$, the entropy coded quantizer bound of Goblick and Holsinger [4], and $R(D)$, the Shannon rate-distortion bound, may be obtained by suitable choices of M and the number L of input range intervals. Both L and M turn out to be fairly small even for rather small distortions. Through the $R_{LM}(D)$ calculation, a methodology and parameters are obtained for trellis encoding. With only four output levels and less effort, coding performance is obtained which is comparable or superior to previous attempts with a continuous output alphabet. These results are the first constructive codes for analog source digitization which achieve better performance than entropy coded quantizers.

INTRODUCTION

Analog (continuous-amplitude) source digitization in theory and practice has always meant quantization. Quantization is rather a restrictive type of source coding in that an input value (scalar or vector) in a certain decision range is always assigned a reproduction value in that range. An optimum quantizer chooses the decision ranges and reproduction values to minimize the distortion subject to a fixed entropy or vice versa. To obtain the minimal rate of the entropy the outputs must be coded (called entropy coding) by either variable length, buffer-instrumented Huffman codes [1] or permutation codes [2]. Theoretically the number of reproduction values becomes infinite in optimum quantization, but, practically speaking, there is negligible loss in performance if the number of output levels is chosen to be large enough. For example, Noll and Zelinski [3] have shown for a Laplacian source with squared-error distortion that the number of levels need be only twice that of a fixed-level minimum mean squared-error quantizer with no entropy constraint. It is well known that optimal quantization achieves a rate that is always above that of the rate-distortion bound. Goblick and Holsinger [4] have shown that for Gaussian sources and squared error distortion that optimal quantization attains a rate only 0.25 bits above the rate-distortion function. Gish and Pierce [5] further showed for general source distributions and the same distortion measure that the minimal rate of quantization is always .255 bits above the Shannon lower bound as the distortion approaches zero. For many sources the Shannon lower bound and the rate-distortion function coincide in a region of low distortion. We shall show subsequently the failure of optimal quantization to reach the performance of the rate-distortion function may be attributed to the rather restrictive mapping from a decision range to a value within that range.

In rate-distortion theory the optimal alphabet (reproduction levels) is infinite for continuous-amplitude sources. Previous attempts at optimal coding of continuous sources (see Dick et al. [6] and Wilson and Lytle [7]) have approximated a continuous output alphabet with 900 discrete values for a tree or trellis coding simulation There is a theorem in Gallager's book [8, Thm. 9.6.2], however, that states the existence of block codes with any given finite number of output letters that achieve performance arbitrarily close to the rate-distortion bound (presumably for some range of distortions not explicitly stated in the theorem). But, to the authors' knowledge, there have been no efforts to find finite alphabet codes for continuous sources (i.e., analog source digitization techniques) that perform better than the optimum quantizer and close to the rate-distortion bound. Perhaps, the reason is that minimum mean-squared error quantization can perform so well for Gaussian sources--within .255 bits of the absolute minimum. Looking beyond Gaussian sources, Noll and Zelinski [3] have found much larger gaps in performance between optimum quantization and optimal coding with the squared-error criterion for the low rate (high distortion) region. In source digitization there is a strong interest in achieving low rates.

We present in this paper a theory and methodology for obtaining any desired coding performance between the optimum quantizer and rate-distortion bounds with an output alphabet of finite size given a stationary, memoryless, continuous-amplitude source and single-letter distortion measure. We accomplish it by the introduction of rate-distortion theory for continuous sources and size constrained output alphabets with rate-distortion functions and associated coding theorems. We then present a convergent algorithm for the evaluation of one of these finite output alphabet rate-distortion functions and associated coding theorems. We then present a convergent algorithm for the evaluation of one of these finite output alphabet rate-distortion functions. We test and apply this theory to the specific case of the Gaussian source and squared-error distortion measure to afford comparison to the well-known quantizer and coding bounds. By calculation of a rate-distortion function we are able to show that almost any performance between the two bounds is attainable for a given size output alphabet if the distortion exceeds that of the corresponding minimum distortion (Max [9]) quantizer. Using the parameters found from the calculation, we have simulated by computer trellis codes at 1 bit per source symbol with performance in the desired range. The trellis search algorithm is the first known use for continuous sources of a modified version of the single stack algorithm proposed by Gallager for symmetric sources and distortion measures [10]. Not only did we find for the first time finite alphabet codes for a continuous source with performance close to the rate-distortion bound, we achieved comparable or better performance with far less effort than a previous attempt by Wilson and Lytle [7] with an approximately continuous output alphabet.

RATE-DISTORTION THEORY FOR FINITE OUTPUT ALPHABETS

In this section, we introduce rate-distortion functions for time-discrete, memoryless sources with size constrained output alphabets. Consider a time-discrete, memoryless continuous alphabet source X which is to be reproduced with the user or output alphabet Y. Define a single letter distortion measure $\rho(x,y)$ between the realizations x of the source X and y of the output Y. Although the development that follows can be straightforwardly generalized to abstract alphabets and probability measures, we shall always assume for the sake of simplicity that the alphabets of X and Y are real numbers and that the source is described by a probability density $p(x)$. Suppose the source is to be represented at the output by a finite set of M letters (real numbers) from Y denoted

$Y_M = \{y_1, y_2, \ldots, y_k, \ldots, y_M\}$. The average mutual information and average distortion between the input and output of this encoding are, respectively,

$$I_M(q,y) = \int_{-\infty}^{\infty} \left\{ \sum_{k=1}^{M} p(x) q_M(y_k/x) \log \frac{q_M(y_k/x)}{Q(k)} \right\} dx \tag{1a}$$

and

$$d_M(q,y) = \int_{-\infty}^{\infty} \left\{ \sum_{k=1}^{M} p(x) q_M(y_k/x)\ \rho(x,y_k) \right\} dx \tag{1b}$$

where $q_M(y_k/x)$ is the conditional probability of y_k given x, called the input-output transition probability, and

$$Q(k) = \int_{-\infty}^{\infty} p(x) q(y_k/x)\, dx \tag{1c}$$

is the probability of the output letter y_k. The quantities q and y in the parenthesis following I_M or d_M denote the dependence on sequences of transition probability functions $q_M(y_k/x)$ and output letters y_k, respectively. All logarithms here and hereafter have base 2 and average mutual informations and rates are in units of bits.

First, we fix the number M of output letters and define the M-letter rate-distortion function $R_M(D)$ as

$$R_M(D) = \inf_{\Gamma_D} I_M(q,y) \tag{2}$$

where Γ_D is the set of all measurable transition probabilities $q_M(y_k/x)$, $k = 1,2,\ldots,M$ and M-letter output alphabets Y_M such that $d_M(q,y) \leq D$. The usual rate-distortion function R(D) results when M is allowed to vary over the set of real numbers. (In Shannon's words, "the set Y_M is varied in both choice and number (size) and the set $q_M(y_k/x)$ is varied over all measurable transition probabilities" [11].) The solution R(D) is generally obtained with a continuous reproduction alphabet. For example, the well-known solution for a real-valued Gaussian source of variance σ^2 and the squared-error distortion measure $\rho(x,y) = (x-y)^2$ is

$$R(D) = \max\left(\frac{1}{2} \log \frac{\sigma^2}{D},\ 0\right) \tag{3}$$

with Y the set of all real numbers.

In the definition of $R_M(D)$, we minimize $I_M(q,y)$ over the set of all measurable transition probabilities $q_M(y_k/x)$ yielding average distortion no larger than D. Instead, let us consider the input values x divided into L disjoint ranges $r_j = [a_{j-1}, a_j)$, $j = 1,2,\ldots,L$ $(a_o = -\infty,\ a_L = +\infty)$ and the piecewise constant transition probabilities

$$q_{LM}(y_k/x) = q_{k/j} \qquad x \varepsilon r_j,\ j = 1,2,\ldots,L \tag{4}$$

for each $k = 1,2,\ldots,M$. When we fix L and minimize $I_M(q,y)$ over all choices of the L ranges and transition probabilities, $q_{k/j}$, subject to the distortion D, we expect the minimal $q_{k/j}$'s to be stepwise approximations to the minimal $q_M(y_k/x)$'s in the $R_M(D)$ problem. In Figure 1, we illustrate the stepwise approximation $q_{k/j}$ to $q_M(y_k/x)$. Of course, the larger the number of ranges L covering the input space of real numbers, the better the approximation.

Let us now formalize the introduction of the piecewise constant test channel and calculate the average mutual information $I_{LM}(q,y,a)$ and distortion $d_{LM}(q,y,a)$ by substituting Eqn. (4) into (1) for a selected partition of the input alphabet into ranges $r_j = [a_{j-1}, a_j)$, $j = 1,2,\ldots,L$. We call the ranges r_j the decision ranges and their endpoints a_j the decision levels and use the notation "r" and "a" for the sequences of decision ranges and levels, respectively. For a given set of decision levels "a," the average mutual information and distortion for the piecewise constant test channel become

$$I_{LM}(q,y,a) = \sum_{j=1}^{L} \int_{a_{j-1}}^{a_j} \sum_{k=1}^{M} p(x) q_{LM}(y_k/x) \log \frac{q_{LM}(y_k/x)}{Q_k} dx \tag{5a}$$

$$= \sum_{j=1}^{L} \sum_{k=1}^{M} p_j \, q_{k/j} \log \frac{Q_{k/j}}{Q_k}$$

$$d_{LM}(q,y,a) = \sum_{j=1}^{L} \int_{a_{j-1}}^{a_j} \sum_{k=1}^{M} p(x) q_{LM}(y_k/x) \rho(x,y_k) \, dx \tag{5b}$$

$$= \sum_{j=1}^{L} \sum_{k=1}^{M} p_j q_{k/j} \, \rho_{LM}(j,k)$$

where we have defined

$$p_j = \int_{a_{j-1}}^{a_j} p(x) \, dx \tag{5c}$$

$$\rho_{LM}(j,k) = \int_{a_{j-1}}^{a_j} p(x) \rho(x,y_k) \, dx / p_j \, . \tag{5d}$$

The quantity p_j is the probability of an input value x falling in the range $r_j = (a_{j-1}, a_j)$ and $\rho_{LM}(j,k)$ is the conditional average distortion introduced by the mapping of a value of x in the range r_j to a value y_k given that x is in r_j. Note that once the decision and output levels a and y are specified, we have a discrete input-discrete output problem with a distortion measure $\rho_{LM}(j,k)$ derived from our original measure $\rho(x,y_k)$. This fact inspires us to define the $R_{LM}^{(a)}$ function by the following minimization. Let

$$R_{LM}^{(a)}(D) = \inf_{\Gamma_D^{(a)}} I_{LM}(q,y,a) \tag{6}$$

where $\Gamma_D^{(a)}$ is the set of all L by M matrices whose elements are the probabilities $q_{k/j}$, M-tuples with real elements y_k, and (L-1)-tuples with real elements $a_1, a_2, \ldots, a_{L-1}$ $(a_0 = -\infty,\ a_L = +\infty)$ such that

$$d_{LM}(q,y,a) \leq D .$$

In shorthand we shall refer to $\Gamma_D^{(a)}$ as the set of triples (q,y,a) satisfying the above distortion constraint. The problem of finding $R_{LM}^{(a)}(D)$ is more complex than that of the conventional discrete input-discrete output R(D) evaluation in that minimal sets of decision and output levels a and y must be found in addition to the minimal test channel q. One simplification that later allows us to formulate a numerical algorithm is to restrict the input ranges to be of equal sizes except for the two end intervals. That is, let

$$a_j - a_{j-1} = \alpha \qquad j = 2,3,\ldots,L-1 \tag{7}$$

Then the dimension of the portion of the minimization belonging to the decision level space is reduced from L-1 to 1. We therefore, formulate a related problem and rate-distortion function as

$$R_{LM}(D) = \inf_{\Gamma_D^{(\alpha)}} I_{LM}(q,y,\alpha) \tag{8}$$

where $\Gamma_D^{(\alpha)}$ is the set of all triples (q,y,α) such that $d_{LM}(q,y,\alpha) \leq D$. $I_{LM}(q,y,\alpha)$ and $d_{LM}(q,y,\alpha)$ are $I_{LM}(q,y,a)$ and $d_{LM}(q,y,a)$, respectively, in Eqn. (5) with the substitution of Eqn. (7). Since $\Gamma_D \supset \Gamma_D^{(a)} \supset \Gamma_D^{(\alpha)}$ and and $Y \supset Y_M$, we deduce immediately the following inequality relationships between the various rate-distortion functions:

$$R_{LM}(D) \geq R_{LM}^{(a)}(D) \geq R_M(D) \geq R(D) \tag{9}$$

CODING THEOREMS

Because we have simplified our original $R_M(D)$ function to the discrete input-discrete output rate-distortion functions, $R_{LM}^{(a)}(D)$ and $R_{LM}(D)$, we see that the usual source coding theorems will apply to these functions for codes which are D-admissible under the single-letter criterion $\rho_{LM}(j,k)$. By D-admissible we mean that the average code distortion is no larger than D. The equivalence between $\rho_{LM}(j,k)$ and the original measure $\rho(x,y_k)$ guarantees D-admissibility under $\rho(x,y_k)$ as well. We therefore state the following source coding theorem and converse whose proofs follow those of the analogous theorems in Berger [12, Thms. 3.2.1, 3.2.2].

Theorem 1

A block $\underline{x}$ of length n from a stationary memoryless (i.i.d) source X is to be encoded into an equal length block $\underline{y}$, whose elements are members of the optimum M letter alphabet Y_M of $R_{LM}^{(a)}(D)$, under a single-letter distortion measure $\rho_{LM}(j,k)$ originating from $\rho(x,y)$ by (5d). Let D_M be the minimum achievable distortion under $\rho_{LM}(j,k)$. Given $D \geq D_M$ and $\varepsilon > 0$, an integer n can be found such that there exists a (D+ε)-admissible code of blocklength n with output alphabet Y_M of rate

$$R_Y = \frac{1}{n} \log N < R^{(a)}_{LM}(D) + \varepsilon$$

where N is the size of the codebook.

Furthermore, no code with an M-letter output alphabet that is D-admissible under distortion measure $\rho_{LM}(j,k)$ has a rate less than $R^{(a)}_{LM}(D)$ for $D \geq D_M$.

As $R_{LM}(D)$ is a special case of $R^{(a)}_{LM}(D)$, the same theorem can be stated for it as well. There is a minimum achievable distortion $D_M > 0$, because we are demanding that a continuous alphabet be represented with only a finite number M of reproduction symbols. This minimum distortion occurs in Eqn. (5b) by letting

$$\rho(x) = \min \rho(x,y_k) \tag{10a}$$

and setting

$$q_M(y_k/x) = 1 \tag{10b}$$

for any k and all associated values of x realizing this minimum [cf. 12, sec. 2.4] and $q_M(y_k/x) = 0$ otherwise. The above transition probability distribution, decision and output levels are those of the Max quantizer [9], which is a deterministic mapping that achieves the minimum possible distortion for a given number of output levels. We state the conclusion in the following theorem.

Theorem 2

The minimum rate for encoding a source with the least possible distortion D_M for a given number M of output levels is the output entropy of the minimum distortion, M-level (Max [9]) quantizer.

Quantization is a means of achieving digitization of a continuous source to a finite number of reproduction levels and to date has been the only one. It is optimal only in rare circumstances, one of which is that $D = D_M$. We shall now investigate the coding procedure for achieving $R^{(a)}_{LM}(D)$ and link quantization to this context.

OPTIMAL SOURCE DIGITIZATION AND QUANTIZATION

An optimal coding procedure (M-level digitization) envisioned from the evaluation of $R^{(a)}_{LM}(D)$ is to determine the L optimal partitions, a, of the input alphabet and the optimal M-letter output alphabet Y_M. In the context of the $R^{(a)}_{LM}(D)$ problem, the optimum quantizer can be characterized by a special form of the piecewise constant test channel of (4) given in (10) as

$$q_{LM}(y_k/x = q_{Q_M}(y_k/x) = \begin{cases} 1 \; , & x\varepsilon r_k = [a_{k-1}, a_k) \\ & \qquad\qquad k = 1,2,\ldots,M \\ 0 \; , & \text{otherwise} \end{cases} \tag{11}$$

The optimum quantizer's transition probability distribution is a mapping from a given range of input values to one value in the same range. (The number of decision ranges L and output levels M are equal, since for $L > M$ we can join contiguous ranges to form one range for the mapping.) As such it is a much cruder approximation to the optimal test channel $q_M(y_k/x)$

than the general piecewise constant channel in (4), as Figures 1 and 2 illustrate. The significant difference in these test channel approximations is that optimal coding, unlike quantization, allows a nonzero probability of a transition from a decision range to an output level belonging to a different range.

Clearly for a fixed M the minimum rate $R^{(a)}_{Q_M}(D)$ with a fixed distortion D for the quantizer Q_M is no smaller than that of $R^{(a)}_{LM}(D)$, i.e., $R^{(a)}_{Q_M}(D) \geq R^{(a)}_{LM}(D)$. Analogously, for range intervals of equal size α, $R^{(\alpha)}_{Q_M}(D) \geq R_{LM}(D)$. The optimum uniform quantizer with distortion D has the rate $R^*(D) = \inf_M R^{(\alpha)}_{Q_M}(D)$, which is the lower envelope to the curves of $R^{(\alpha)}_{Q_M}(D)$ for all values of M. For example, Goblick and Holsinger [4] found for the unit variance Gaussian source with squared error distortion that

$$R^*(D) = \frac{1}{4} + \frac{1}{2}\log\frac{1}{D}\ , \tag{12}$$

only .25 bits greater than the rate-distortion function in (3). The lower envelope of the $R_{LM}(D)$ curves must therefore lie on or below $R^*(D)$. Any one $R_{LM}(D)$ curve shares one point at $D = D_M$ with $R^{(\alpha)}_{Q_M}(D)$ and, as we shall soon see, falls below $R^*(D)$ and approaches $R(D)$ very rapidly for $D > D_M$.

AN ALGORITHM FOR $R_{LM}(D)$

An algorithm is now introduced for the calculation of $R_{LM}(D)$ for any source distribution $p(x)$ and single letter distortion measure $\rho(x,y)$. The algorithm will then be applied specifically to the case of the Gaussian source with $\rho(x,y) = (x-y)^2$. By the definition of $R_{LM}(D)$ in (8), we first select L and M, the numbers of equal size input decision ranges and output levels respectively. Then we must find the decision range size α, the output levels y, and ML transition probabilities $q_{k/j}$ (Eqn. (3)) such that $I_{LM}(q,y,\alpha)$ is minimized under the condition that $d(q,y,\alpha) \leq D$. Let us first reduce the dimension of the problem by assuming some α and seek a rate-distortion function for that α denoted by $R_{LM}(D_s,\alpha)$. We select an initial set of output levels $y°$. Then by Eqn. (5) we have the usual discrete input-discrete output problem with known input and output alphabets and probabilities and a well-defined single-letter distortion measure $\rho_{LM}(j,k) = \rho^{\circ}_{LM}(j,k)$. A numerical solution for the minimal $q^{\circ}_{k/j}$'s and rate-distortion function of the source with distortion measure $\rho_{LM}(j,k)$ is obtained with the algorithm of Blahut [13]. The calculation is initiated by fixing a parameter $s < 0$, which generates a point $(D°, R°)$ and the minimal transition probabilities $q^{\circ}_{k/j}$ for the initial output set $y°$ and given α. As the rate $R_o = I_{LM}(q,y,a)$ in Eqn. (5) with the minimal transitional probabilities $q^{\circ}_{k/j}$ and the given decision range interval α, it is not a function of the output levels, just their probabilities. So we can vary the output levels with R_O fixed to obtain a minimum of $D°$. Now that we have a new set of output levels, which we call y^1_k, we have a new distortion measure $\rho^1_{LM}(j,k)$ as seen by (5d). So we repeat the Blahut algorithm for this new distortion measure for the output levels y^1_k (and the same s) to obtain the corresponding point (D^1,R^1) on the rate-distortion curve. We now minimize D^1 by varying y^1_k to obtain a new set of output levels y^2_k and corresponding distortion measure $\rho^2_{LM}(j,k)$. Another iteration of the Blahut algorithm for these levels and distortion measure produces the point (D^2,R^2). We now see the basis of an algorithm which converges to what we call $R_{LM}(D_s,\alpha)$, the minimum rate for a fixed α and a distortion D_s generated by the parameter s. We now restate the algorithm for $R_{LM}(D_s,\alpha)$.

$R_{LM}(D_s,\alpha)$ Algorithm :

1. Fix $s < 0$.
2. Fix α.
3. Specify set of output levels $y^0 = (y^0_1,\dots,y^0_M)^t$; set $i=0$.
4. Calculate distortion measure $\rho^i_{LM}(j,k)$ as given in (4).
5. Using the Blahut algorithm find the point (D^i,R^i) on the rate-distortion curve and the minimizing transition probabilities $q^i_{k/j}$.
6. Calculate a new set y^{i+1} of output levels such that the distortion
$$\tilde{D}^i = \sum_{j=1}^{L}\sum_{k=1}^{M} q^i_{k/j} \int_{a_{j-1}}^{a_j} p(x)\rho(x,y^{i+1}_k)\,dx$$
is a minimum of D_i.
7. Using some stopping criterion (e.g., $D^i-\tilde{D}^i < \varepsilon$) either go to 8 if criterion is met or set $i = i+1$ and go to 4.
8. $R_{LM}(D_s,\alpha) = R^i$, $D = D^i$, $q_{k/j} = q^i_{k/j}$, $y = y^i$

A proof of the convergence of the algorithm to $R_{LM}(D)$ and a computer program listing appear in [14].

$R_{LM}(D)$ FOR A GAUSSIAN SOURCE WITH SQUARED-ERROR DISTORTION

We shall now present results for the calculation of $R_{LM}(D)$ in the case of the unit variance Gaussian source with $\rho(x,y) = (x-y)^2$. This case affords us the opportunity to compare $R_{LM}(D)$ with two well-known results: Shannon's R(D) in Eqn. (3) and the entropy coded uniform quantizer of Goblick and Holsinger [4], given by R*(D) in Eqn. (12). In Figure 3 we show $R_{LM}(D)$ for $M = 8$ and $L = 8,16,32$, and 64 compared to R(D) and R*(D). First, we observe that the larger L is, the closer $R_{LM}(D)$ is to R(D) for a given D. This result is in agreement with our statements that as L, the number of ranges covering the input space, approaches infinity, $R_{LM}(D)$ approaches $R_M(D)$ which is arbitrarily close to R(D). Secondly, all $R_{LM}(D)$ curves start at D_M, where we have $R_{LM}(D_M)$ equal to the Max quantizer entropy (Thm. 2), and then rapidly fall below $R^*(D)$ as D increases. The larger the value of L, the smaller the value of D such that $R_{LM}(D) < R^*(D)$. In Fig. 4 we display the effect of varying M with L fixed. We plot $R_{LM}(D)$ for $L = 64$ and $M = 8, 16$, and 32 along with $R^*(D)$ and $R(D)$. Each $R_{LM}(D)$ curve has its own minimum distortion point D_M where $R_{LM}(D_M)$ is the Max quantizer entropy for the given number of output levels. As M increases, D_M decreases and the closer $R_{LM}(D)$ is to $R(D)$ for a given D. A very important conclusion from Figs. 3 and 4 is that L and M do not have to be increased indefinitely to obtain performance close to the ideal $R(D)$. For example, for $D = .09773$, $R_{64,8}(D) = 1.687$ and $R(D) = 1.6775$, a difference of .0095 bits. Only a very small decrease in rate can be achieved by increasing either L or M.

TRELLIS ENCODING SIMULATION AND RESULTS

In order to test and apply the preceding theory, we undertake a computer simulation of the trellis encoding of an i.i.d. Gaussian source at a rate of 1 bit per source symbol for the squared error distortion measure. Viterbi and Omura [15] have shown that for memoryless sources trellis codes can perform arbitrarily close to the rate-distortion bound. The branches of the trellis are populated with symbols of the output alphabet drawn independently and at random from the probability distribution of the output symbols. The output symbols and their probabilities are obtained from the calculation of $R_{LM}(D)$. The input range size α is also a product of the $R_{LM}(D)$ calculation and is used also with the output symbols to calculate the distortion matrix with values $\rho_{LM}(j,k)$ as given in (5d).

A calculation of $R_{LM}(D)$ for $L = 64$ and $M = 4$ at an appropriately chosen value of s yielded a rate of $R_{LM} = 1.00023$ bits per symbol and $D = 0.25287$. We also calculated for another value of s, $L = 64$ and $M = 4$, $R_{LM} = .99995$ and $D = .25104$. These distortions are only 0.05 and 0.02dB, respectively, above the rate-distortion bound of $D = .25$ at $R = 1$ bit per source symbol. Since the calculated rate is very close to the desired encoding rate of 1 bit per symbol, we decided to use these output levels and their probabilities and the distortions from these $R_{LM}(D)$ calculations for the trellis codings with both $M = 64$ and $M = 4$.

The trellis search algorithm is chosen to be a variation of the single stack algorithm first proposed and analyzed by Gallager [10] for sources and distortion measures obeying certain symmetry conditions. The justification for applying this or any other tree or trellis coding algorithm to asymmetric sources and measures is provided by the theorems proved by Davis and Hellman [16]. As far as the authors are aware, this algorithm has not been used previously for encoding continuous sources.

We present now the results of the trellis code simulations for a rate of 1 bit per source symbol. We intend to compare results for $L = 64$, $M = 4$ and $L = 64$, $M = 64$. In every case 10 blocks of 250 successive source outputs are encoded with a trellis of 512 states (nodes per stage). The population of the trellis branches with output symbols is undertaken in the spirit of random coding with complementarity. The idea of complementarity, used by Anderson [17] for binary i.i.d. sources with Hamming distortion measure, is that every generation of an output symbol is followed by its complement. It achieves better performance with less storage requirements. The target distortion is chosen to be $\varepsilon = .05$ above that of $R_{LM}(D)$ bounds calculated previously.

Table 1

Average Results of One Bit Trellis
Encoding of Unit Variance Gaussian Source*

M	$\langle D\rangle$	$\langle B\rangle$	G(dB)
4	.289 (.013)	12.24 (3.80)	0.88
64	.282 (.009)	17.35 (5.1)	0.98

*10 data blocks of 250 symbols each

The results are summarized in Table 1 above where we present the average distortion $\langle D\rangle$ and number of branches searched per source symbol $\langle B\rangle$, and the distortion performance gain over optimum quantization G. The numbers in parentheses are the standard deviations of the preceding estimates. For $M = 4$, $G = 0.88$dB and for $M = 64$, $G = 0.98$dB, compared to the absolute theoretical maximum of 1.53dB. We have achieved digitization with 4 of 64 output levels that performs better than any quantizer and close to the rate-distortion bound. Consistent with the theory, the $M = 64$ codes perform only 0.1dB better than those for $M = 4$ but at the expense of forty-two percent more branches searched per source symbol and more computer storage. The search effort, characterized by $\langle B\rangle$, and the performance of these simulations can be compared to the simulation results of Wilson and Lytle [7] at 1 bit per source symbol. They used the same source and distortion measure in searching a trellis with the M- and Viterbi algorithms. They populated the branches randomly with a 900 discrete level approximation to the normally distributed continuous output letters. Their best result, obtained with the M algorithm, was a distortion of .286 (graph-interpolated) with 128 branches searched per source symbol (64 paths retained per node). By comparison, with $M = 4$ and 64, we obtain distortions of .289 and .282 with 12.24 and 17.35 branches searched per source symbol. Therefore we achieve either performance almost as good for $M = 4$ and better for $M = 64$ with far less search effort and much smaller output alphabets in both cases. These results are a consequence of both the $R_{LM}(D)$ theory and the trellis search algorithm.

SUMMARY AND CONCLUSIONS

We have formulated a new theory of analog source digitization by introduction of the finite output alphabet rate-distortion function $R_{LM}(D)$. We have tested this theory in the particular case of the memoryless Gaussian source with squared-error distortion measure. By suitable choices of L and M, which need not be large, the value of $R_{LM}(D)$ may be chosen to be between the optimum quantizer bound $R^*(D)$ and the rate-distortion function $R(D)$ for any $D > D_M$, the minimum possible distortion of an M level quantizer. The coding theorems establish the significance of $R_{LM}(D)$ as an approachable coding bound. The calculation of $R_{LM}(D)$ furnishes a set of output levels, output level probabilities, and an input partition size α, which are used for a trellis coding simulation. The results of the simulation with a Gaussian i.i.d. source and squared-error distortion measure provide codes of rate 1 bit per symbol with distortion close to that predicted by the theory. They are the first finite alphabet codes with performance better than $R^*(D)$ and close to $R(D)$. They are obtained with a variation of Gallager's single stack algorithm and represent the first use of this algorithm with continuous sources. The search effort is far smaller than a similar attempt achieving comparable results with the M-algorithm for an essentially continuous output alphabet.

REFERENCES

1. D. A. Huffman, "A method for the construction of minimum redundancy codes," Proc. IRE, Vol. 40, pp. 1098-1101, Sept. 1952.
2. T. Berger, "Optimum quantizers and permutation codes," IEEE Trans. Inform. Theory, Vol. IT-18 pp. 759-765, Nov. 1972.
3. P. Noll and R. Zelinski, "Bounds on quantizer performance in the low bit-rate region," IEEE Trans. Comm., Vol. COM-26, pp. 300-304, Feb.1978.
4. T. J. Goblick, Jr. and J. L. Holsinger, "Analog source digitization: A comparison of theory and practice," IEEE Trans. Inform. Theory, Vol. IT-13, pp. 323-326, April 1967.
5. H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," IEEE Trans. Inform. Theory, Vol. IT-14, pp. 676-683, Sept. 1968.
6. R. J. Dick, T. Berger, and F. Jelinek, "Tree encoding of Gaussian sources," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 332-336, May 1974.
7. S. G. Wilson and D. W. Lytle, "Trellis encoding of continuous-amplitude memoryless sources," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 404-409, May 1977.
8. R. G. Gallager, Information Theory and Reliable Communications, New York: J. Wiley & Sons, 1968.
9. J. Max, "Quantizing for minimum distortion," IRE Trans. Inform. Theory, Vol. IT-6, pp. 7-12, March 1960.
10. R. G. Gallager, "Tree encoding for symmetric sources with a distortion measure," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 65-76, Jan. 1974.
11. C. E. Shannon and W. Weaver, The Mathematical Theory of Communications, Urbana, IL: University of Illinois, 1949.
12. T. Berger, Rate Distortion Theory, Englewood Cliffs, NJ: Prentice-Hall, 1971.
13. R. E. Blahut, "Computation of channel capacity and rate-distortion functions," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 460-473, July 1972.
14. W. A. Finamore, Optimal Encoding of Continuous-Amplitude Sources with Constrained Size Output Alphabet, Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI, May 1978.
15. A. J. Viterbi and J. K. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 325-332, May 1974.
16. C. R. Davis and M. E. Hellman, "On tree coding with a fidelity criterion," IEEE Trans. Inform. Theory, Vol. IT-21, pp. 373-378, July 1975.
17. J. B. Anderson, Algorithms for Tree Coding with a Fidelity Criterion, Ph.D. Dissertation, School of Electrical Engineering, Cornell University, Ithaca, NY, May 1972.

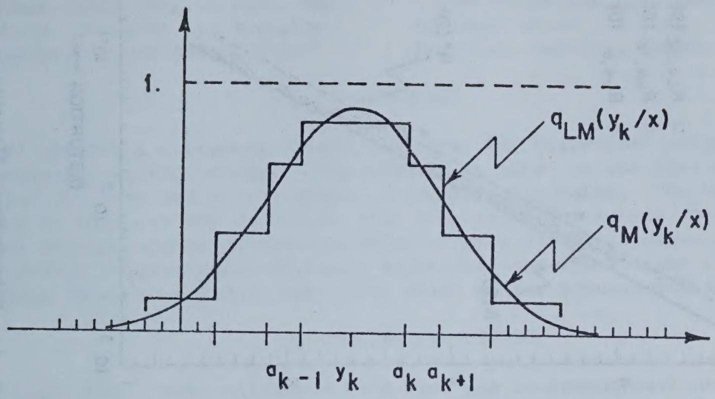

1.
$q_{LM}(y_k/x)$
$q_M(y_k/x)$
a_{k-1}
y_k
a_k
a_{k+1}

Fig. 1. Comparison of transition probabilities q_M and q_{LM}.

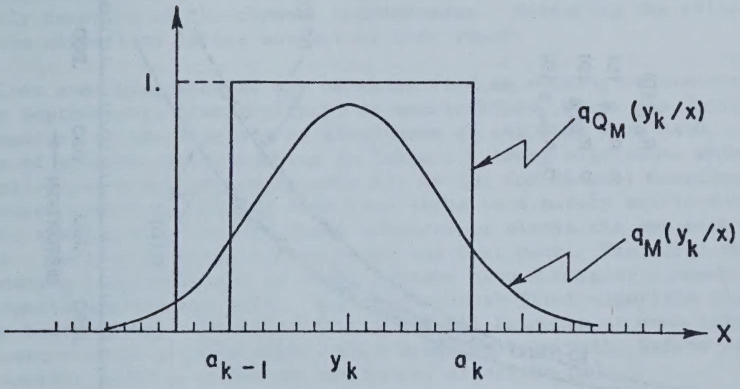

1.
$q_{Q_M}(y_k/x)$
$q_M(y_k/x)$
X
a_{k-1}
y_k
a_k

Fig. 2. Quantizer and test channel transition probabilities to level y_k.

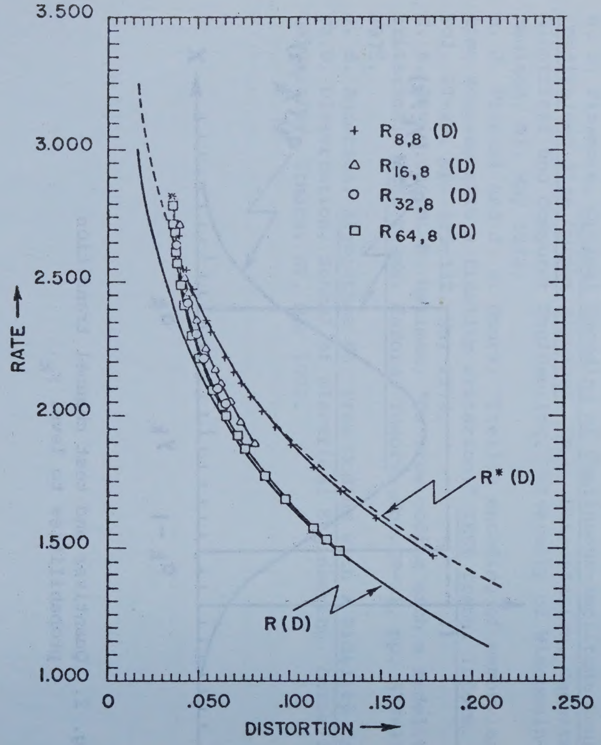


Fig. 3. Rate versus distortion for L = 64, 32, 16, and 8 and M = 8.

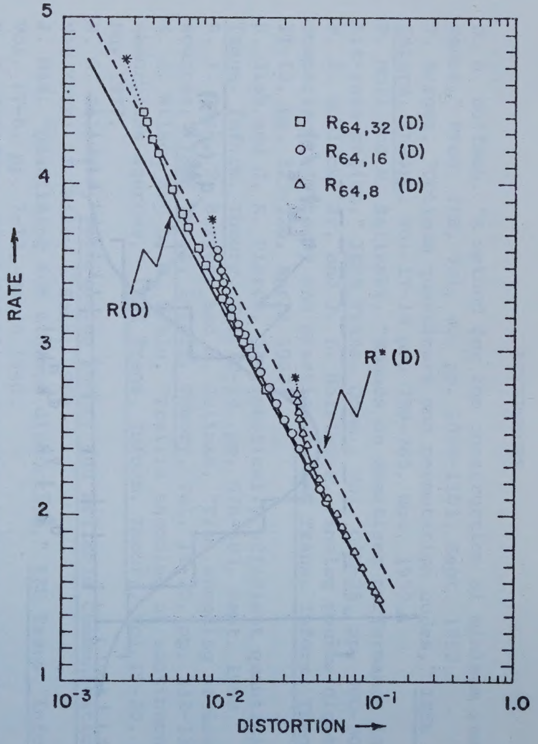


Fig. 4. Rate versus distortion for L = 64 and M = 32, 16, and 8.

A COST FUNCTION FOR SEQUENTIAL CODING

J.B. ANDERSON
Visiting Assoc. Professor
Dept. Elec. Eng. & Comp. Sci.
Univ. of Calif. at Berkeley
Berkeley, California 94720

S. MOHAN
Dept. Elec. Eng. &
Communications Res. Lab.
McMaster Univ.
Hamilton, Ontario, Canada

ABSTRACT

We present a systematic measure of cost for sequential coding algorithms and evaluate certain schemes. The measure is based on the size of storage an algorithm uses and on the number of accesses it makes. The three dominant factors in cost are the number of code tree paths the scheme stores, their length, and the number of tree branches it must view to achieve a performance level. Experimental evidence shows that the lowest cost algorithms are those which do not sort and those which do not proceed metric-first.

1. SEQUENTIAL CODING

Codes with a tree structure find wide use in communications engineering. By a tree code is meant a code whose words may be graphed on a perfectly regular tree structure having σ symbols on each branch and b branches out of each node. Each word corresponds to a path through the tree made up of these branches. The most widely used waveform speech digitizers, including PCM, delta modulation, and differential PCM, utilize tree codes. Convolutional error correcting codes have a tree structure. Intersymbol interference caused by channel band limiting can be viewed as real number convolutional coding, and equalization as sequential decoding. During the encoding process in the first of these and during the decoding process in the last two, it would be best to view all the branches in the code tree, but this is a practical impossibility. Instead, a search algorithm is employed which views some but not others in a predetermined fashion, attempting to find good digitization of the speech waveform or a likely decoding of the channel transmission. Measuring the efficiency of these algorithms is the subject of this paper.

Code searching schemes may be classified as sorting or non-sorting, and as depth-first, breadth-first, or metric-first, where the "metric" is some measure of the fidelity or likelihood of the code tree path. A number of schemes are summarized in Table I. Among algorithms which sort, the well-known stack algorithm (see [1] or [3] for channel decoding or [2] for source encoding) extends code tree paths in a purely metric-first manner, meaning that the next path extended is always the one with the best metric. Sorting is used to single out the best path. The usual method is an ordering procedure, but we shall mention also a merging procedure which has significantly lower cost. A purely breadth-first algorithm that sorts is the M-algorithm. This algorithm views all branches at once that it will ever view at a given depth, then drops certain paths before continuing on. Another sorting scheme is the bucket algorithm [1].

A second class of algorithms does not sort; that is, paths are never compared with one another. The simplest such method is the single stack algorithm, a purely depth-first method suggested by Gallager [4]. This scheme simply pursues a path until its metric falls below a discard criterion, and at any one time it stores the identity of only one path. A direct implementation is a single push-down stack. (It should be mentioned that the more widely known "stack algorithm" cited above in fact contains no stack, but only a list.) A familiar variation of the single stack method is the Fano algorithm; the peculiar character of this search stems from its method of computing the discard criterion. A more sophisticated non-sorting procedure is to set aside certain good paths for later attention as they appear in the depth-first search. An example is the 2-cycle algorithm [5].

All of the above methods are selective search algorithms, which leave some, and usually most, paths unviewed. The Viterbi algorithm [6] is an exhaustive search, which considers all possibilities inherent in the code by exhibiting all the states of the code generation structure. The Viterbi algorithm takes advantage of the trellis structure of a code, a structure possessed by anypractically useful tree code.

The efficiency of sequential coding algorithms has traditionally been measured by the number of branches that must be visited to achieve a given level of source encoder fidelity or decoder probability of error. When divided by the branches released as output, this is called the node computation. It has become increasingly clear that the node computation alone is not a sufficient measure of cost. Several authors ([8], [9]) have found that stack algorithm source encoding is exceptionally time consuming, even though it has the least node computation of any known method. Experience with M-algorithm hardware speech encoders [10] shows that this method is efficient despite a poor node computation.

In sequential channel decoding and in equalization, the situation is similarly confused. The Viterbi algorithm finds wide use despite its exhaustive nature. The selectively searching algorithms which should be more efficient suffer erasures brought on by computation overload, and find only occasional use in applications like deep space communication. What seems to be missing here are factors to account for the size and complexity of the information structures peculiar to each algorithm, and the difficulties inherent in their use.

2. ALGORITHM COST FUNCTIONS

A more realistic measure of algorithm cost is based on the number of storage elements in a scheme and the number of accesses to them. For any algorithm, one can define the space cost to be the size of resources that must be reserved for its use and the time cost to be the number of accesses to this resource. We shall consider the product of these two, the space x time cost, as our primary measure of a scheme's total cost.

A space x time product cost measure is most suited to hardware as opposed to software implementations. Storeage and other blocks are assumed to have a purchase cost proportional to their speed and their number; that is, buying a block at twice the cost that will sustain twice the accesses per second is no cost advantage, nor is a block of twice the size at twice the cost. The product measure also assumes that components "wear out" to a significant degree. If they did not, infinitely many accesses could be purchased at finite cost, and the product cost per unit of production would be zero.

A second measure of cost, more suited to software implementations in a multiprogrammed environment, is the space + time cost. This measure stresses more the opportunity cost foregone by assigning resources to a user. If a coding algorithm consumes a given number of central processor cycles, these cannot be assigned to someone else, and similarly the storage allocation in central memory is denied to a third user. Both incomes are

<table>
<tr><th></th><th>Metric-First</th><th>Breadth-First</th><th>Depth-First</th></tr>
<tr><td>Sorting</td><td>Stack Alg.
Merge Alg.
Bucket Alg.
(roughly)</td><td>M-Alg.</td><td></td></tr>
<tr><td></td><td colspan="2">--Haccoon's Alg. (both)--</td><td></td></tr>
<tr><td></td><td colspan="3">xx← Mult. Stack Alg. →xx</td></tr>
<tr><td>Non-Sorting</td><td></td><td></td><td>Single Stack Alg.
Fano Alg.</td></tr>
<tr><td></td><td></td><td colspan="2">--2-Cycle Alg. (both)--</td></tr>
</table>

TABLE 1: Search rationales for certain selective search algorithms.

Algorithm	(Space)(Time)	(Space)+(Time)
Single Stack	$\sim LE[C_{SS}]$	$\sim L+E[C_{SS}]$
Fano	$\sim LE[C_{FA}]$	$\sim L+E[C_{FA}]$
2-Cycle	$\sim LE[C_{2C}]$	$\sim LL_{pd}+E[C_{2C}]$
Stack	$\sim LS^2E[C_{SA}]$	$\sim LS+SE[C_{SA}]$
Merge	$\sim L(S^{4/3}E[C_{SA}]+S^2)$	$\sim LS+(S+S^{2/3}E[C_{SA}])$
Bucket	$\sim L(SE[C_{SA}]+S^2)+H$	$\sim LS+(S+E[C_{SA}])+H$
M-	$\sim LS^2\mathrm{Log}S$	$\sim LS+S\log S$

TABLE 2: Asymptotic cost of certain algorithms in the limit of intensive searching, per output symbol released. Space in b-ary symbols, Time in accesses to storage. C = branches visited/output symbol released, L = length of retained paths, S = number of retained paths.

separately foregone, so that the proper charge is the sum of the two. Most central computer installations use such a charging scheme, with appropriate constants multiplying the two components.

For either cost measure, different constants of proportionality affect space and time, but we will be most interested in the asymptotic cost of an algorithm as the coding task becomes increasingly difficult, and here the constants have little significance. Parallel processing is asymptotically of no benefit under the space x time measure, since there is no gain in trading space for time.

Other measures could be proposed than the space and time costs of storage blocks, but we have chosen this measure because it relates closely to the nature of coding algorithms. Code search algorithms basically move in and out of storage data about a large stock of code tree paths. The chief challenge in their design is to store path data where it can be easily found and to choose paths in a way that will shorten the coding process. Once a path is obtained, visiting the branches forward of it is not difficult. Such tasks as updating path metrics, checking for ambiguous output symbols, and generating code word letters, can generally be viewed as a multiplier close to unity on the cost of storage accesses.

Another measure of cost applicable to the algorithms which sort is the number of comparisons executed during the sorting. This measure does not account for both space and time, and as mentioned previously, not all algorithms sort.

3. MAJOR DETERMINANTS OF COST

Three variables dominate the asymptotic cost of the algorithms we have analyzed, and we list them at the outset. These are the length of path an algorithm can retain, L, the number of paths it can retain, S, and the expected node computation defined before, E[C]. We assume that L and S are finite and fixed in value. It is important to realize that algorithms differ significantly when these are so constrained; when the number of paths exceeds S, for instance, some mechanism must delete excess paths, and whenever a path exceeds length L, its oldest branch must be checked to insure that it is consistent with other paths kept to this depth. These routines may change the asymptotic cost.

In 4 we review the space and time requirements for a number of algorithms; detailed material which is devoted to describing the algorithms is available in [7]. It is not our intention to optimize over the choice of the three major cost factors, or the numerous lesser factors in the cost. An accurate optimization and a comparison of the algorithms to choose an "optimal" one are huge tasks. However, the experimental work thus far available does allow certain speculations, and these appear in 5.

Results are summarized in Table 2. It is clear that there are dramatic differences in algorithms; all depend linearly on L, but some, like the single stack algorithm, have no dependence on S, while others range as high as $S^2\log S$. As has been shown in earlier work, E[C] varies widely, and the M-algorithm in particular has no separate dependence on this factor.

4. COST FUNCTION EVALUATIONS

It will be useful to keep in mind the ultimate aims of the coding process, as well as certain actions that all algorithms perform.

A source encoder searches for a path with distortion as small as possible. For probabilistic sources, the limit to such activity is Shannon's distortion-rate function $\Delta(R)$, where $R = \log_2 b/\sigma$ is the output rate of the code tree. For real-life sources like speech, $\Delta(\ )$ is often not computable, but this is of little concern to the encoder. A channel decoder searches for a path with likelihood as high as possible; it will find the transmitted path correctly in finite expected node computation whenever the channel data rate falls below R_{comp}.

For either encoding or decoding, one group of source or channel outputs arrives for each tree depth. The searching process begins at a root node and continues until some path reaches depth L. The algorithm then decides once and for all which first branch to release as output; the end node of this branch becomes the new root node. Searching resumes until some path again exceeds L branches. The procedure continues indefinitely in this "sliding block" fashion, releasing some branch at depth ℓ and accepting a new data group at depth $\ell + L$.

Paths are described by path maps made up of b-ary symbols, $\{0,\ldots,b-1\}$, one for each branch. The code word letters on a path's terminal branch are somehow computable from its path map. Associated with each map is a metric μ, either a likelihood of the path or a measure of its distortion, and sometimes an indication of the path's length or pointers to other storage locations. Metric functions are discussed in [7] and elsewhere.

The following specific actions are performed in the same manner by all algorithms.

Extend path: The algorithm extends a path one branch forward, "viewing" sometimes one branch, sometimes all b branches out of the end node. Viewing includes calculating the code word symbols that belong on the branch, fetching the input data corresponding to its tree depth, calculating the increment to the path metric caused by the branch, and forming the new metric total.

Ambiguity check: The algorithm checks all eldest path map symbols to determine of they are consistent with the symbol released as output. If not, the path must be deleted.

Delete Path: The algorithm deletes an entire path map. A deletion must occur whenever the number of paths stored exceeds S, and whenever a path fails an ambiguity check.

Release Output Symbol: The algorithm releases as output the earliest symbol of the best path map it has. In sorting algorithms, this triggers an ambiguity check to make certain that all path maps have the same symbol at this depth.

We begin with two non-sorting algorithms. A distinguishing feature of non-sorting schemes is that they store only a single path, extending or backtracking along it in response to the value of the path metric.

A. The Single Stack Algorithm

This algorithm proceeds depth-first directly through the code tree until the path metric falls below a discard criterion B; the search then backtracks to the first untried branch and proceeds depth-first again. The symbols of the path map are stored in a push-down stack, and in the steady-state operation of the algorithm, an output path map symbol is forced out the bottom whenever a depth is visited for the first time.

By convention we assume that on first visiting a node, branch 0 out of the node is viewed, on the second visit branch 1, and so on until all branches are viewed, at which time the search must backtrack. An example of this routine appears in Fig. 1 for $b = 2$; X's indicate a path that has fallen below the discard criterion.

Using a single push-down stack and a comparison to a discard criterion, the single stack algorithm is the simplest of all search algorithms. The space cost is simply L b-ary symbols for the push-down stack (plus a small overhead for side registers and code letter generation); the time cost is overbounded by 2 accesses/branch viewed. The space x time cost is thus

$$\simeq L\ E\ [C_{SS}] \text{ access-symbols/output branch} \qquad (1)$$

where C_{SS} denotes the node computation of the single stack algorithm, and here and throughout $X \simeq Y$ means that $\log X/\log Y$ tends asymptotically to 1. The space + time cost is $\simeq L + E\ [C_{SS}]$.

The Fano algorithm is a variant of the single stack procedure in which the discard criterion varies up and down as a function of the path metric. Searching proceeds depth-first until a path falls below B, but B is raised in increments whenever possible during visits to new nodes, and is lowered when necessary during returns to previously visited nodes. The product cost of the Fano algorithm is again of the form $\simeq L\ E\ [C_{FA}]$, but it is not clear which of C_{FA} and C_{SS} is the larger. The opportunistic changing of B undoubtedly reduces the set of nodes visited, but the algorithm may visit certain of the nodes many times. Nonetheless, if C_{FA} can be measured, then the form (1) gives the total cost.

B. The 2-Cycle Algorithm

Algorithms similar to the single stack procedure search depth-first, backtracking only when the path falls below the discard criterion. Another method which uses the same stack structure is to search all paths lying above the criterion and having length $\ell \leq L$. Of these, only a much smaller set of "good" paths, say those with metric $\mu \geq A$, are saved for later attention. This procedure has been called the 2-cycle algorithm. Its product cost has the same form as (1), although its sum cost has the more complicated form shown in Table 2 (L_{pd} is the number of saved good paths). The precise formulation of the 2-cycle algorithm is complicated; further details may be found in [5] or [7].

The remaining algorithms sort, meaning that they compare paths on the basis of metric in order to decide which to extend and which to delete. These algorithms view fewer branches than non-sorting algorithms, but the cost of sorting is often very high.

C. The Stack Algorithm

As mentioned previously, the stack algorithm is based not on a stack, but on a list of code tree paths. The usual view is that this is an ordered list; the next path extended is always the best in terms of μ, and sufficient worst paths are deleted to keep the list at size S paths.

Since a path's metric indicates the likelihood that the best path in the code tree lies ahead of it, it comes as no surprise that this metric-first procedure appears to find a path at a given metric level with the least node computation of any scheme (see [1] or [2]). Despite this, the space-time cost of the stack algorithm seems to exceed that of any other scheme; there are other metric-first procedures having much lower cost, but these remain expensive.

Most of the cost of the stack algorithm lies in ordering the path list. To insert a new path requires about S accesses to the path list, counting the accesses needed to delete excess paths, while the storage size is about $S(L + \log L + p)$ b-ary symbols. Here, log L represents the symbols that specify each path's length and p the path's metric, both of which the algorithm requires. The algorithm needs an ambiguity check, but this cost is of lower order [7].

Total product cost is thus about $\simeq S^2L\ E\ [C_{SA}]$ access-symbols per output branch, neglecting the lower order terms p and log L, where C_{SA} is the node computation of the stack algorithm. Unless $E\ [C_{SA}]$ is very small, the added S^2 factor will make this cost much larger than that of the non-sorting algorithms. The sum cost is about $LS + SE\ [C_{SA}]$, which is a factor S larger than the non-sorting algorithms.

D. The Merge Algorithm

The cost of the stack algorithm can be greatly reduced while still retaining the strictly metric-first feature if two ordered lists are jointly maintained. The strategy will be to order new paths into a short auxiliary list and occasionally merge the list into the main list. We have called such a procedure the merge algorithm [7], [11].

It can be shown [7] that the merge procedure reduces the comparisons required to insert a new path from order S to order $S^{1/2}$, and that the extra storage needed is asymptotically insignificant. This translates into the sum and product costs given in Table 2.

Other procedures using side list have been given by Chevillat and Costello [13] and by Haccoon and Ferguson [14].

E. The Bucket Algorithm

Another strategy to reduce sorting effort is to retreat to a roughly metric-first search from an approximate ordering. Jelinek [1] proposed that the range of μ be quantized into segments called "buckets" and incoming paths be sorted only into the correct bucket. He called this procedure the bucket algorithm.

The bucket algorithm is subtle in its operation and quite complex to describe. The reader must again suffer being referred to [7]. Nonetheless, accesses and storage sizes may again be counted in the manner above, with the results given in Table 2. The H term accounts for the cost of specifying into which bucket a path should be placed. Setting aside this term, one can see another large improvement in the order of S in the cost expressions. An S factor remains, however, unlike the non-sorting algorithms which have none at all.

F. The M-Algorithm

We conclude with a breadth-first sorting algorithm. In general, such a procedure views all branches at depth ℓ that it will ever view, deletes paths according to some criterion, and then moves on to the next depth. Unlike the search of Fig. 1, it never backtracks. The M-algorithm deletes all paths except a fixed number M. Breadth-first searches are synchronous (that is, all paths have the same length) and they are effective at low intensities of searching; they are thus good candidates for practical application [10], [12].

In its specific operation, the M-algorithm moves forward by extending the M paths it has retained to form bM new paths. All the terminal branches are compared to the input data corresponding to this depth, metrics computed, and the (b-1)M poorest paths deleted. The heart of the algorithm is the sorting procedure which deletes these paths, a job that amounts to sorting a randomly ordered list. Optimal schemes such as Mergesort and Heapsort are available to do this in order M log M comparisons.

Letting $S = M$, we can calculate the space cost to be of order LS, with S log S accesses required for each branch released as output. The product cost is $\simeq LS^2 \log S$ and the sum cost is $\simeq LS + S \log S$. The dependence on S is of high order, but there is no E [C] term.

5. AN OPTIMAL ALGORITHM: EXPERIEMENTAL EVIDENCE

As mentioned in 3., finding the optimal settings for L, S, and E [C] which combine to give the least cost is a formidable task. But much experimentation has been done, and some naive concludsions are possible by inserting experimentally determined values in the forms of Table 2. It must be stressed that these forms are asymptotic, and only large differences will be significant. The results generally favor algorithms that do not sort, and point away from metric-first searching.

We consider first sequential source encoding. Table 3 reproduces simulations done by the authors for certain algorithms used with random tree codes. The source is the binary letter i.i.d. source with "Hamming" distortion measure (unit penalty for mismatched letters, zero penalty otherwise), the encoder rate is 1/2 output bit/source bit, and all the algorithms achieve an average distortion of about 0.125, 15% above the distortion-rate function.

It is clear the two non-sorting algorithms have greatly reduced cost compared to the stack or M-algorithms under either the product or sum cost evaluation. Even the merge and bucket algorithms (borrowing the same L, S, and E [C] as the stack algorithm) fall well short, and in fact perform only as well as the M-algorithm.

Turning now to sequential channel decoding, we consider the case of rate 1/2 convolutional codes and a binary symmetric channel with crossover probabilities in the range 0.02 to 0.045. The 0.045 error rate corresponds to $R_{comp} = 1/2$; for higher error rates, the node computation lacks a first moment. Jelinek ([1], p. 680 ff) reports $E[C_{SA}]$ increasing from 2 up to about 4 through this error range, while Chevillat and Costello ([13], pg. 1469) report an E [C] of about 2.8 at error probability 0.025 for their multiple stack algorithm. The storage factors S reported by both lie in the range 1000 - 3000. The Fano algorithm, on the other hand, requires up to seven times as much node computation as the stack schemes.

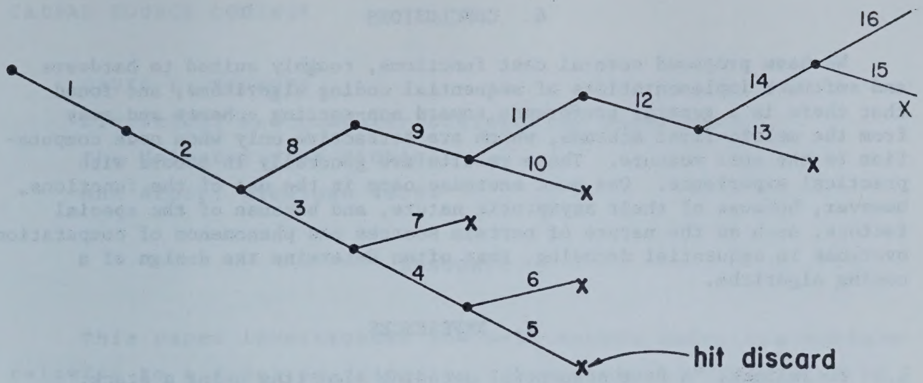


FIG. 1: Push-down stack search, b = 2. Downward branch (0-th) taken first, then upward branch (1-st); numbers show order of visiting, X means path metric hits discard criterion.

	Branches Viewed E[C]	Paths Stored S	Space×Time Cost	Space+Time Cost
Stack	200	>500	10^{10}**	200K
M-	500	250	$7\text{X}10^7$	51K
2-Cycle	1000	1*	200K	8500
Single Stack	1500	1	300K	1700

L = 200-300, all cases (200 used for cost).

* About 150 paths of average length 50 were kept in the save stack.

** $2\text{X}10^8$ with Merge Alg.; $7\text{X}10^7$+H with Bucket Alg.

TABLE 3: Evaluation of cost for certain algorithms, taken from experimental data. Binary i.i.d. source with Hamming distortion, R = 1/2, encoded distortion 0.125 (Shannon limit = 0.110).

Nonetheless, if we assume that all algorithms have about the same L, the cost evaluation of the Fano algorithm is less by a factor of 10^3.

It is of interest to compare the cost of these methods with that of the Viterbi algorithm, using a code which achieves about the same decoded error probability (10^{-5} - 10^{-4}). This we have done elsewhere [7], with the result that the Viterbi algorithm cost lies in between the sorting and non-sorting algorithms. At lower decoded error probabilities, the non-sorting algorithms become cheaper.

6. CONCLUSIONS

We have proposed several cost functions, roughly suited to hardware and software implementations of sequential coding algorithms, and found that there is a general preference toward non-sorting schemes and away from the metric-first schemes, which are attractive only when node computation is the cost measure. These results are generally in accord with practical experience. One must exercise care in the use of the functions, however, because of their asymptotic nature, and because of the special factors, such as the nature of certain sources and phenomenon of computation overload in sequential decoding, that often determine the design of a coding algorithm.

7. REFERENCES

1. F. Jelinek, "A fast sequential decoding algorithm using a stack," IBM J. Res. Develop. vol. 13, pp. 675-685, November 1975.
2. J.B. Anderson, "A stack algorithm for source coding with a fidelity criterion," IEEE Trans. Information Theory, vol. IT-20, pp. 211-266, March 1974.
3. K. Sh. Zigangirov, "Some sequential decoding procedures," Probl. Peredach. Inform., vol. 2, no. 4, pp. 13-25, 1966.
4. R.G. Gallager, "Tree encoding for symmetric sources with a distortion measure," IEEE Trans. Information Theory, vol. IT-20, pp. 65-76, January 1974.
5. J.B. Anderson and F. Jelinek, "A 2-cycle algorithm for source coding with a fidelity criterion," IEEE Trans. Information Theory, vol. IT-19, pp. 77-92, January 1973.
6. A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," IEEE Trans. Information Theory, vol. IT-13, pp. 260-269, April 1967.
7. J.B. Anderson and S. Mohan, "A systematic analysis of cost for sequential coding algorithms," in submission, IEEE Trans. Information Theory, July 1978.
8. R.J. Dick, T. Berger, and F. Jelinek, "Tree encoding of Gaussian sources," IEEE Trans. Information Theory, vol. IT-20, pp. 332-336, May 1974.
9. S. Mohan, Ph.D. Research, 1977.
10. J.B. Anderson and C.-W. P. Ho, "Architecture and construction of a hardward sequential encoder for speech," IEEE Trans. Commun., vol. COM-25, pp. 703-707, July 1977.
11. S. Mohan and J.B. Anderson, "Stack algorithm speech encoding," 1977 IEEE International Symposium on Information Theory, Ithaca, New York, 10-14 October 1977.
12. S.G. Wilson, "Tree encoding of speech at 8000 bps," IEEE Symposium, ibid.
13. P.R. Chevillat and D.J. Costello, "A multiple stack algorithm for erasurefree decoding of convolution codes," IEEE Trans. Communications, vol. COM-25, pp. 1460-1470, December 1977.
14. D. Haccon and M.J. Ferguson, "Generalized stack algorithms for decoding convolutional codes," IEEE Trans. Information Theory, vol, IT-21, pp. 638-651, November 1975.

CAUSAL SOURCE CODING*

DAVID L. NEUHOFF
Electrical and Computer Engineering Department
The University of Michigan
Ann Arbor, Michigan 48104

ABSTRACT

This paper investigates how well source codes can perform relative to a fidelity criterion, when they are restricted to be causal in an appropriate sense. The principal result is that for memoryless sources, the best performance is achieved either by blocklength one codes or by time-sharing of blocklength one codes. The performance of the latter technique can be evaluated straightforwardly.

INTRODUCTION

Source coding theorems for block, sliding-block, tree and trellis source codes have shown that these classes contain codes that achieve the rate distortion function R(D), which is the optimum performance theoretically attainable relative to some distortion measure. Although these classes contain some codes that are causal in a sense to be defined shortly, it is widely believed that it is the noncausal codes in these classes, and only the noncausal coes, that achieve R(D). Nevertheless there are causal codes of great practical interest. Indeed for some sources there are causal codes whose performance comes quite close to R(D). In other cases there are causal codes that are believed to perform quite well but have not yielded to rigorous analysis.

In this paper we present quantitative results on the optimum performance theoretically attainable by causal codes for memoryless sources. The goal is to discover how much is lost relative to R(D) by the restriction to causal codes, or, equivalently, how much is to be gained by noncausality.

---

* This work was supported by NSF Grant 76-82531.

CAUSAL CODES

We will say that a source code is causal if the reproduction produced by the code of the present source output depends only on the past as well as the present source outputs. (It is not required that the compressed representation of the source output be produced causally.) Thus quantizers, delta modulators, differential pulse code modulators, and adaptive versions of these are all causal, and, for example, uniform quantizers with entropy coding have been shown [1] to come within 1/4 bit of R(D) for Gaussian memoryless sources and small squared-error distortion.

Specifically, we consider two kinds of causal codes. Let $\{X_k\} = \{X_k; k=-\infty,\ldots,\infty\}$ be a stationary source with alphabet A. A stationary causal code is characterized by a function $f^s : A^0_{-\infty} \to A$. The reproduction of the source output X_k is $\hat{X}_k = f^s(X^k_{-\infty})$, where X^n_m denotes $(X_m \ldots, X_n)$. The average distortion produced by the code relative to a distortion measure $\rho(x,\hat{x})$ is

$$\rho(f^s) \triangleq E\ (X_0, \hat{X}_0) = E\rho(X_0, f^s(X^0_{-\infty}))$$

and the rate of the code is

$$r(f^s) \triangleq H(\hat{X}) \triangleq \lim_{n\to\infty} \frac{1}{n} H(\hat{X}_1, \ldots, \hat{X}_n) \qquad (1)$$

Stationary causal codes are sliding-block codes, and may be approximated by causal codes with finite window length.

A block causal code with blocklength N is characterized by a set of N functions $f^b_n : A^n_1 \to A$, $n=1,\ldots,N$. The reproduction of X_k produced by the code is $\hat{X}_k = f^b_j(X^k_{k-j+1})$, if $j = k \bmod(N)+1$. We let $f^b(x^N_1) \triangleq (f^b_1(x_1), \ldots, f^b_N(x^N))$ and use f^b to denote the code. The average distortion is

$$\rho(f^b) = E\ \frac{1}{N}\sum_{i=1}^{N} \rho(X_i, \hat{X}_i) = \frac{1}{N}\sum_{i=1}^{N} E\ \rho(X_i, f^b_i(X^i_1))$$

and the rate is again given by (1).

Let us note that the rate of these codes has been measured with entropy. One may obtain an actual compressed representation at the code rate by following the code with a noiseless source code.

We now let $r^s_c(D)$, $r^b_c(N,D)$, $r^b_c(D)$ denote, respectively, the least rate of any stationary causal code, of any blocklength N block causal code, and of any block causal code with average distortion D. Let $\bar{r}_c^b(N,D)$ denote the lower convex hull to $r_c^b(N,D)$. The principle result of this paper is:

Theorem For a stationary, memoryless source

$$r_c^s(D) \geqq r_c^L(D) = \bar{r}_c^b(1,D) \triangleq r_c(D)$$

To prove this theorem we first establish the following lower bound.

$$r_c^s(D) \geqq r_c^L(D) \triangleq \inf H(\hat{X}_0 | X_{-\infty}^{-1}),$$

where the infimum is over all stationary causal codes f^s with $\rho(f^s) \geqq D$. This bound dominates the bound found by Lloyd [2] and in most cases is strictly larger. It follows from the fact that for any f^s, $\hat{X}_{-\infty}^{-1}$ is a function of $X_{-\infty}^{-1}$. Therefore

$$r(f^s) = H(\hat{X}_0 | \hat{X}_{-\infty}^{-1}) \geqq H(\hat{X}_0 | X_{-\infty}^{-1}).$$

This bound, like Lloyd's, can be evaluated by linear programming techniques when A is finite.

We now show $r_c^L(D) \geqq \bar{r}_c^b(1,D)$. Let f^s be any stationary causal code. For any $x_{-\infty}^{-1}$, let $f_{x_{-\infty}^{-1}}^s : A \to A$ be defined according to $f_{x_{-\infty}^{-1}}^s(x_0) = f^s(x_{-\infty}^0)$. Since $\{X_k\}$ is memoryless,

$$H(\hat{X}_0 | x_{-\infty}^{-1}) = H(f^s(X_{-\infty}^0) \mid x_{-\infty}^{-1}) = H(f_{x_{-\infty}^{-1}}^s(X_0)). \tag{2}$$

The function $f_{x_{-\infty}^{-1}}^s$ is a blocklength one code; hence

$$H(f_{x_{-\infty}^{-1}}^s(X_0)) \geqq r_c^b(1,\rho(f_{x_{-\infty}^{-1}}^s)) \geqq \bar{r}_c^b(1,\rho(f_{x_{-\infty}^{-1}}^s)) \tag{3}$$

We now employ (2) and (3) in the definition of $H(\hat{X}_0 | X_{-\infty}^{-1})$ and find

$$H(\hat{X}_0 \mid X_{-\infty}^{-1}) = H(f^s(X_{-\infty}^0) \mid X_{-\infty}^{-1}) = \int H(f^s(X_{-\infty}^0)\ x_{-\infty}^{-1})\, d\mu(x_{-\infty}^{-1})$$

$$\geqq \int \bar{r}_c^b(1,\rho(f_{x_{-\infty}^{-1}}^s))\, d\mu(x_{-\infty}^{-1}) \quad ,$$

where μ denotes the source measure on $A_{-\infty}^{-1}$. Since $\bar{r}_c^b(1,D)$ is convex $\cup$ and since

$$\int \rho(f_{x_{-\infty}^{-1}}^s)\, d\mu(x_{-\infty}^{-1}) = \rho(f^s) \quad ,$$

Jensen's inequality implies

$$H(\hat{X}_0 | X_{-\infty}^{-1}) \geqq \bar{r}_c^b(1,\rho(f^s)) \quad .$$

Taking the infimum of both sides over all f^s with $\rho(f^s) \le D$ proves $r_c^L(D) \ge \bar{r}_c^b(1,D)$.

Next we will use an argument, similar to one used in (3), to show that given a block causal code it is possible to create a stationary causal code with nearly identical performance. It will follow that $r_c^b(D) \ge r_c^s(D)$.

The idea is to create a stationary code which, in effect, uses the block code most, but not all, of the time. This is done by first producing a blocking sequence $\{S_n\}$ such that most of the time $\{S_n\}$ counts repeatedly through the integers one through N, where N is the blocklength of the code. Then whenever $S_{n+1}^{n+N} = (1,2,\dots,N)$, the block code is used to code X_{n+1}^{n+N}. In particular if $f^b = \{f_1^b,\dots,f_N^b\}$ characterizes the block code, then we define a stationary causal code by

$$\hat{X}_n = f^s(X_{-\infty}^n) = f^b_{s_n}(X^n_{n-s_n+1}). \tag{4}$$

The fact that $\{S_n\}$ counts repeatedly through the integers implies that most of the time f^s uses f^b over and over again. There will be occasional glitches in the blocking sequence, but these will not disturb the performance significantly; indeed they are necessary in order that f^s be stationary. Of course $\{S_n\}$ must be produced causally. The existence of a suitable blocking sequence is demonstrated by the following result, which is proved in the Appendix.

Lemma: Let $\{X_n\}$ be a stationary memoryless source. Then for any N,K,δ there is a stationary causal encoder $g: A^0_{-\infty} \to \{1,2,\dots,N\}$, such that if $S_n \triangleq g(X^n_{-\infty})$, then

(i) $\Pr(S_n = S_{n-1} \oplus 1) \ge 1-\delta$,

where $\oplus$ denotes addition modulo N,

(ii) $\frac{1-\delta N^2}{N} \le \Pr(S_n = j) \le \frac{1+\delta N^2}{N}$, $j=1,\dots,N$

(iii) S_1^L is independent of X_1^L

(iv) $\frac{1}{L} H(S_1^L) \le \frac{1}{L}(h(\delta') + (1+L\delta')\log N)$,

where $L \triangleq KN$, $\delta' = KN^2\delta$, and $h(\alpha) = -p\log p$

-(1-p)log(1-p). (All logarithms will be base 2).

We may now apply the lemma with N equal to the blocklength of the code and for some δ and K to be chosen later. With f^s defined as in (4) the average distortion will be

$$\begin{aligned}\rho(f^s) &= \sum_{i=1}^{N} E[\rho(X_0, f^s(X_{-\infty}^0)) \,|\, S_0=i] \Pr(S_0=i) \\ &\leqq \sum_{i=1}^{N} E\rho(X_0, f_i^b(X_{-i+1}^0)) \, \frac{1+N^2\delta}{N} \\ &= (1+N^2\delta)\rho(f^b)\end{aligned}$$

If δ is chosen small enough then

$$\rho(f^s) \leqq \rho(f^b) + \varepsilon$$

The rate of f^s may be bounded as follows.

$$\begin{aligned}r(f^s) = H(\hat{X}) &\leqq \frac{1}{L}H(\hat{X}_1^L) \leqq \frac{1}{L}H(\hat{X}_1^L, S_1^L) \\ &\leqq \frac{1}{L}H(S_1^L) + \frac{1}{L}H(\hat{X}_1^L | S_1^L) \qquad (5)\end{aligned}$$

The first term on the right hand side of (5) is bounded by condition (ii) of the Lemma. Consider next the second term. If $S_1^L = s_1^L$ and s_1^L counts repeatedly through the integer 1 to N starting with $s_1 = j-1$, then

$$\begin{aligned}H(\hat{X}_1^L | s_1^L) \leqq\ & H(\hat{X}_1^{N-j} | s_1^L) + \sum_{k=1}^{K} H(\hat{X}_{kN-j+1}^{kN-j+N} | s_1^L) \\ & + H(\hat{X}_{L-j+1}^{L} | s_1^L) \qquad (6)\end{aligned}$$

The sum of the first and last terms in (6) is at most NlogQ, where Q is the number of distinct code letters produced by f^b. For k =1,..,K, $\hat{X}_{kN-j+1}^{kN-j+N}$ is obtained by block coding X_{kN-j+1}^{kN-j+N} using f^b. Hence

$$H(\hat{X}_{kN-j+1}^{kN-j+N} | s_1^L) \leqq Nr(f^b) \qquad (7)$$

After substituting (6), (7), and condition (iii) of the Lemma into (5) we find

$$r(f^s) = \frac{1}{KN}\left(h(KN^2\delta) + (1+K^2N^3\delta)\log N\right) + \frac{1}{K}\log Q + r(f^b)$$

If K is chosen large enough and δ small enough, then

$$r(f^s) \leqq r(f^b) + \varepsilon .$$

Since f^b and ε are arbitrary, it follows that

$$r_c^s(D) \leqq r_c^b(D).$$

Finally we claim that $r_c^b(D)$ is convex $\cup$. This follows from a time sharing argument. Given some δ, $0 \leqq \lambda \leqq 1$ and two block causal codes with blocklengths N_1 and N_2 that achieve $r_c^b(D_1)$ and $r_c^b(D_2)$, respectively, one may construct a new block causal code with blocklength N, $N \geqq N_1$ and $N \geqq N_2$, that uses the first code on the first λN source symbols and the second code on the last $(1-\lambda)N$ source symbols. This code will have distortion $\lambda D_1 + (1-\lambda)D_2$, and if N is sufficiently large the rate will approach $\lambda r_c^b(D_1) + (1-\lambda) r_c^b(D_2)$ which by definition must be no smaller that $r_c^b(\lambda D_1 + (1-\lambda)D_2)$. Hence $r_c^b(D)$ is convex $\cup$.

Let us collect the results. We have shown

$$r_c^b(D) \geqq r_c^s(D) \geqq r_c^L(D) \geqq \bar{r}_c^b(1,D)$$

and that $r_c^b(D)$ is convex $\cup$. Since by definition $r_c^b(D) \leqq r_c^b(1,D)$, it follows that $r_c^b(D) = \bar{r}_c^b(1,D)$, and so

$$r_c^b(D) = r_c^s(D) = r_c^L(D) = \bar{r}_c^b(1,D).$$

THE EVALUATION OF $r_c(D)$

For finite alphabet memoryless sources $r_c^L(D)$ may be evaluated by linear programming techniques. Alternatively, one may plot the performance of each blocklength 1 code ; $\bar{r}_c^b(1,D)$ is the lower convex hull of these points. Figures 1 and 2 present two simple examples.

For continuous alphabet sources, blocklength one causal codes are more commonly called quantizers. Our result shows that the best performance achievable with causal codes for memoryless sources is the lower convex hull to the best performance of quantizers with entropy coding. This curve is ordinarily convex, for example if $\{X_k\}$ is Gaussian [1]. Therefore, in this case quantizers are the best causal source codes.

APPENDIX

Proof of Lemma: Given N, K, δ, let $L = NK$ and let $a^0_{-p} \in A^0_{-p}$ be a sequence such that

$$\Pr(X^n_{n-p} = a^0_{-p}) \leq \delta$$

We will say that a "glitch" occurs at time n if $X^n_{n-p} = a^0_{-p}$ and define a glitch process $\{G_n\}$ according to

$$G_n = \begin{cases} 1 , & \text{if } x^{n-L}_{n-L-p} = a^0_{-p} \\ 0 , & \text{otherwise} \end{cases} .$$

The blocking sequence $\{S_n\}$ is then created as follows. If n-t is the time of the most recent glitch at or before time n, then $S_n = t \bmod(N) + 1$; i.e.,

$$S_n = g(X^n_{-\infty}) \triangleq t \bmod(N) + 1,$$

if $t \geq 0$, $G_{n-t} = 1$, and $G_{n-u} = 0$ for all $0 \leq u < t$. With this definition of g , it is possible for S_n to differ from $S_{n-1} \oplus 1$ only if a glitch occurs at time n-L . Hence

$$\Pr(S_n \neq S_{n-1} \oplus 1) \leq \Pr(G_n = 1) = \Pr(X^n_{n-p} = a^0_{-p}) \leq \delta \quad .$$

This proves (i). Statement (ii) follows from (i), and statement (iii) follows from the facts that S_1^L is a function of $G^L_{-\infty}$, $G^L_{-\infty}$ is a function of $X^0_{-\infty}$, and the latter is independent of X_1^L . We now prove (iv). If δ is very small, then with high probability S_1^L consists of K complete cycles of the integers 1 through N, beginning with some integer j. Define

$$p_j = \Pr(S_1^L = (j, j+1, \ldots, N, 1, \ldots, N, \ldots, 1, \ldots, j-1)) .$$

Then

$$\sum_{j=1}^{N} p_j \geq \Pr(S_n = S_{n-1} \oplus 1, \ n = 2, \ldots, L) \geq 1 - L\delta \ .$$

But $p_j \leq 1/N$, all j, since the relative frequency of the event $S_1^L = \{(j, \ldots, N, \ldots, 1, \ldots, j-1\}$ can be at most 1/N. Therefore for any j

$$p_j = \sum_{k=1}^{N} p_k - \sum_{k \neq j} p_k \geq 1 - L\delta - (N-1)\frac{1}{N} \geq \frac{1 - LN\delta}{N} \triangleq \frac{1 - \delta'}{N} \ .$$

The entropy of S_1^L is cleary upper bounded by the entropy of the distribution on $\{1,\ldots,N\}^L$ having N probabilities equal to $(1 + \delta')/N$ and N^L-N probabilities equal to $\delta'/(N^L-N)$. Hence

$$\frac{1}{L}H(S_1^L) \leq \frac{1}{L}\left(-N\frac{1-\delta'}{N}\log\frac{1-\delta'}{N} - (N^L-N)\frac{\delta'}{N^L-N}\log\frac{\delta'}{N^L-N}\right)$$

$$= \frac{1}{L}\left(h(\delta') + (1-\delta')\log N + \delta'\log(N^L-N)\right)$$

$$\leq \frac{1}{L}\left(h(\delta') + (1+L\delta')\log N\right),$$

and this proves (iv).

REFERENCES

[1] T.J. Goblick Jr. and J.L. Holsinger, "Analog Source Digitization: A Comparison of Theory and Practice," IEEE Trans. on Inform. Theory, Vol. IT-13, April 1967, pp. 323-326.

[2] S.P. Lloyd, "Rate vs. Fidelity for the Binary Source", BSTJ, Vol. 56, March 1977, pp. 427-437.

[3] P.C. Shields and D.L. Neuhoff, " Block and Sliding-Block Source Coding", IEEE Trans. on Inform. Theory, Vol. IT-23, March 1977, pp. 311-215.

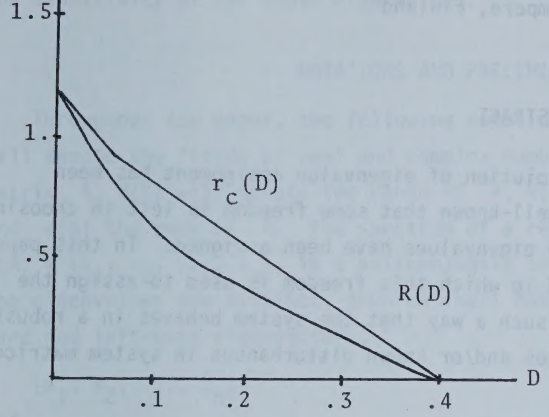


Figure 1 : A ternary memoryless source; $p_1=.6$, $p_2=.34$, and $p_3=.06$.

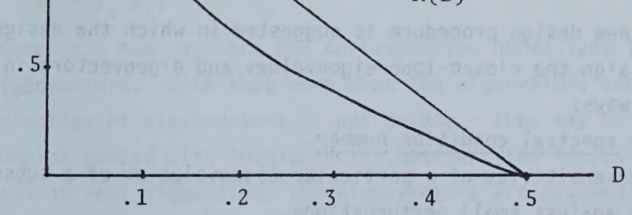


Figure 2 : A ternary memoryless source; $p_1=.5$, $p_2=.25$, and $p_3=.25$.

ON ROBUST EIGENVALUE/EIGENVECTOR ASSIGNMENT

SEPPO POHJOLAINEN HEIKKI KOIVO
Tampere University of Technology
Tampere, Finland

ABSTRACT

For linear systems the solution of eigenvalue assignment has been thoroughly studied. It is well-known that some freedom is left in choosing the feedback matrix once the eigenvalues have been assigned. In this paper a new procedure is suggested in which this freedom is used to assign the closed-loop eigenvectors in such a way that the system behaves in a robust way against small disturbances and/or known disturbances in system matrices.

INTRODUCTION

For linear systems the solution of eigenvalue assignment has been throughly studied [1]. It is well-known that some freedom is left for a multi-input, multi-output system when eigenvalues have been assigned. This freedom can be used to design the system in such a way that it behaves in a robust way against (small) perturbations in system matrices.

Some relevant work in this field has already been done. Ramar and Gourishankar [2] minimize the sensitivity coefficients of the system. Their approach is limited to optimization and is not an algebraic method. Shah, Fisher and Seborg [3] have reviewed most of the work done in the area and proposed that the closed loop eigenvectors should be assigned to make some eigenvalues invariant to partly known perturbations is system parameters.

In this paper, a new design procedure is suggested in which the design freedom is used to assign the closed-loop eigenvalues and eigenvectors in one of the following ways:

1. Minimizing the spectral condition number.
2. Minimizing the sensitivity of a particular eigenvalue or of a subset of eigenvalues against small perturbations.

The design idea applies both to the case when the perturbation of system matrices is totally unknown and when it is only partly unknown.

The design method presented here is different from [2] and [3] and makes it possible to combine the previous ideas so that it is possible - if the perturbation is partly known - to reduce its effect totally on some eigenvalues and then to use the remaining design freedom to reduce the sensitivity of the other eigenvalues.

NOTATIONS AND PRELIMINARIES

Throughout the paper, the following notation will be used: $\mathbb{R}$ and $\mathbb{C}$ will denote the fields of real and complex numbers, respectively. For a matrix A, $R(A)$ will denote the range of A, $N(A)$ the nullspace of A, and $r(A)$ the rank of A. The spectrum of a real matrix A, $\sigma(A) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$, is a selfconjugate set of its eigenvalues. If the eigenvalues are distinct, then we shall have a complete set of right-hand and left-hand eigenvectors,

$$\{x_1, x_2, \ldots, x_n\}$$

and

$$\{y_1, y_2, \ldots, y_n\},$$

respectively. These eigenvectors form a bi-orthogonal system, that is,

$$y_i^T x_j = 0, \quad i \neq j, \quad i,j = 1, \ldots, n,$$

$$y_i^T x_i \neq 0, \quad i = 1, \ldots, n.$$

The orthogonality of two vectors v, w is denoted by $v \perp w$. In what follows, we shall assume that the eigenvectors are normalized to unity $\|x_i\| = \|y_i\| = 1$, $i = 1, \ldots, n$, where $\|\cdot\|$ stands for Euclidean norm. The corresponding matrix norm is given by $\|A\| = \max\limits_{\|x\| \leq 1} \|Ax\|$.

Now we may define the condition number for an eigenvalue λ_i as

$$s_i = y_i^T x_i, \tag{1}$$

where y_i and x_i are the corresponding normalized left and right eigenvectors. Note that even when the eigenvalues are distinct, the selection of eigenvectors is not unique - they may be multiplied by a complex number with modulus unity,however, the $|s_i|$:s are unique. In the case of real eigenvalues the eigenvectors are assumed to be real. Clearly, when the eigenvalues are distinct

$$0 \neq |s_i| = |y_i^T x_i| \leq \|y_i^T\| \|x_i\| = 1,$$

for all $i = 1, \ldots, n$.

The following result is well-known. Let A have distinct eigenvalues, and let

$$X = (x_1, x_2, \ldots, x_n) \tag{2}$$

and

$$Y = (y_1/s_1, y_2/s_2, \ldots, y_n/s_n) \tag{3}$$

be matrices whose columns consist of the normalized right and left eigenvectors corresponding to eigenvalues λ_i, $i = 1, \ldots, n$. Then we have

$$Y^T AX = \lceil \lambda_1, \lambda_2, \ldots, \lambda_n \rfloor, \tag{4}$$

where $\lceil\ \rfloor$ stands for a diagonal matrix. In addition $Y^T = X^{-1}$.

PERTURBATION RESULTS FOR SIMPLE EIGENVALUES

This section is based on chapter 2 of Wilkinson [4].

Let $A \in \mathbb{R}^{nxn}$ and $G \in \mathbb{R}^{nxn}$ be real matrices, scaled so that the elements $|a_{ij}| \leq 1$ and $|g_{ij}| \leq 1$, $i,j = 1, \ldots, n$. Let the spectrum of A, $\sigma(A) = \{\lambda_1, \ldots, \lambda_n\}$, consist of distinct eigenvalues and let $\{x_1, x_2, \ldots, x_n\}$ and $\{y_1, y_2, \ldots, y_n\}$ be the corresponding sets of normalized right and left eigenvectors. Then the eigenvalues $\lambda_i(\varepsilon)$ and right eigenvectors $x_i(\varepsilon)$ of the perturbed matrix $A + \varepsilon G$, where $\varepsilon \in \mathbb{R}$ and $|\varepsilon|$ is a small number, are given by convergent power series

$$\lambda_i(\varepsilon) = \lambda_i + k_{1i} \cdot \varepsilon + k_{2i}\varepsilon^2 + \ldots, \tag{5}$$

$$x_i(\varepsilon) = x_i + \varepsilon z_{1i} + \varepsilon^2 z_{2i} + \ldots, \quad i = 1, \ldots, n. \tag{6}$$

If $|\varepsilon|$ is sufficiently small the main perturbation of λ_i is $\varepsilon \cdot k_{1i}$, where k_{1i} is given by

$$k_{1i} = \frac{y_i^T G x_i}{y_i^T x_i} = \frac{\gamma_{ii}}{s_i}, \tag{7}$$

where we have adopted the standing notation

$$\gamma_{ij} = y_i^T G x_j. \tag{8}$$

So we see that the sensitivity of the eigenvalue against small unknown perturbations is primarily dependent on s_i. Unfortunately an $|s_i|$ may be very small.

First order perturbation term in the perturbation of an eigenvector, say x_1, is given by

$$z_{11} = \frac{\gamma_{21}\ x_2}{(\lambda_1-\lambda_2)s_2} + \frac{\gamma_{31}\ x_3}{(\lambda_1-\lambda_3)s_3} + \dots + \frac{\gamma_{n1}\ x_n}{(\lambda_1-\lambda_n)s_n} \tag{9}$$

Notice that all the other s_i's except s_1 appear in eq. (9) and the factors $(\lambda_1-\lambda_i)$ in the denominators. We may expect the eigenvector corresponding to the simple eigenvalue λ_1 to be comparatively insensitive to small perturbations in A, if the eigenvalues are well-separated and none of the $|s_i|$:s, $i = 2, 3, \dots, n$ are small.

One drawback of the presented argument is that we assumed the perturbation to be small. However, we may write the matrix $A + \varepsilon G$ in a form, where the effect of the perturbation is easily seen, without assuming $|\varepsilon|$ to be small. Let X and Y be as in (2) and (3). Then

$$Y^T(A+\varepsilon G)X = \lceil \lambda_1, \lambda_2, \dots, \lambda_n \rfloor + \varepsilon \cdot \begin{bmatrix} \gamma_{11}/s_1, & \gamma_{12}/s_1, & \dots, & \gamma_{1n}/s_1 \\ \gamma_{21}/s_2, & \gamma_{22}/s_2, & \dots, & \gamma_{2n}/s_2 \\ \\ \gamma_{n1}/s_n, & \gamma_{n2}/s_n, & \dots, & \gamma_{nn}/s_n \end{bmatrix} \tag{10}$$

for all $\varepsilon \in \mathbb{R}$.

A possible measure for the overall sensitivity of eigenvalues of matrix A is given by the spectral condition number. It is well known ([4]) that

$$|\lambda_i-\lambda| \leq \varepsilon \cdot \kappa(X)\ \|G\| \tag{11}$$

for at least one value of i, where λ_i is an eigenvalue of A, λ is an eigenvalue of $A + \varepsilon G$, and the spectral condition number $\kappa(X)$ is defined by

$$\kappa(X) = \|X^{-1}\|\ \|X\|, \tag{12}$$

where $X^{-1} = Y^T$ and X are matrices (2) and (3).*) Note that we do not reguire $|\varepsilon|$ to be small in (10) and (11). The spectral condition number gives a bound for the coefficients s_i, because

$$|s_i^{-1}| \leq \kappa(X), \quad i = 1, 2, \dots, n. \tag{13}$$

---

*) Note that in general the spectral condition number of A is defined to be the smallest value of $\kappa(X)$ for all matrices X, for which $X^{-1}AX$ is diagonal. The simple definition, given above, is sufficient for our purposes.

In any case we have

$$\kappa(X) = \|X^{-1}\| \|X\| \geq \|X^{-1}X\| = 1.$$

For a normal A, $\kappa(X) = 1$, and hence the eigenvalue problem is always well-conditioned, though this is not necessarily true of the eigenvector problem.

ON THE FREEDOM OF EIGENVALUE AND EIGENVECTOR ASSIGNMENT

Consider linear system

$$\dot{x} = Ax + Bu$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $r(B) = m$, $n \geq m$, and the pair (A,B) is controllable. By applying linear state feedback $u = Fx$ we obtain the closed loop system

$$\dot{x} = (A+BF)x. \tag{14}$$

We shall now examine the assignment of closed loop eigenvalues and eigenvectors.

It is well known that F may be chosen to yield any self-conjugate set of closed loop eigenvalues. Unless $m = 1$ the specification of closed loop eigenvalues does not uniquely define F. The design freedom available beyond eigenvalue selection is that of choosing one of the possible closed loop eigenvector sets.

Let $S_\lambda = [\lambda I-A \ B]$ and let

$$K_\lambda = \begin{bmatrix} N_\lambda \\ M_\lambda \end{bmatrix}$$

be a compatibly partitioned matrix, columns of which constitute a basis for $N(S_\lambda)$. It is quite easy to see that the columns of N_λ are linearly independent and $r(N_\lambda) = m$, because B has linearly independent columns and (A,B) is controllable. Also we see that $N_{\bar{\lambda}} = \overline{N}_\lambda$.

The following theorem, due to Moore [5], gives us a way to select eigenvectors for the closed loop system and to calculate the feedback matrix F.

Theorem 1. Let $\{\lambda_i, i = 1, 2, \ldots, n\}$ be a self-conjugate set of distinct complex numbers. There exists a matrix F of real numbers such that $(A+BF)x_i = \lambda_i x_i$, $i = 1, 2, \ldots, n$, if and only if the following three conditions are satisfied for $i = 1, 2, \ldots, n$.

1) Vectors x_i are linearly independent vectors in $\mathbb{C}^n$

2) $x_i = \overline{x_j}$, whenever $\lambda_i = \overline{\lambda_i}$

3) $x_i \in R(N_{\lambda_i})$.

If F exists and $r(B) = m$, then F is unique.

If all the eigenvalues are real numbers, then F is given by

$$F = [w_1, w_2, \dots, w_n] [x_1, x_2, \dots, x_n]^{-1} \qquad (15)$$

where $x_i = N_{\lambda_i} k_i$ for some vector $k_i \in \mathbb{R}^m$ $(\mathbb{C}^m)$ and $w_i = -M_{\lambda_i} k_i$

If a pair of eigenvalues is complex, say $\lambda_1 = \overline{\lambda}_2$, the matrix F is given by

$$F = [w_{1R}, w_{1I}, W] [x_{1R}, x_{1I}, V]^{-1}, \qquad (16)$$

where x_{1R}, x_{1I} are the real and imaginary parts of the eigenvector $x_.$ corresponding to the eigenvalue λ_1. w_{1R} and w_{1I} are the real and imaginary parts of w_1, respectively. The columns of matrices V and W are x_i, $i = 3, 4, \dots, n$ and w_i, $i = 3, 4, \dots, n$, respectively. The procedure can obviously be applied for all complex pairs of eigenvalues.

ON ROBUST EIGENVALUE/EIGENVECTOR ASSIGNMENT

We have seen that the eigenvalues and eigenvectors are quite stable against small perturbations, if the eigenvalues are well separated and the coefficients $|s_i|$, $i = 1, \dots, n$ are not small. Because our system is controllable we may easily separate the eigenvalues. Now the objective is to use the freedom - found by Moore [5] - to assign the closed loop eigenvectors correctly to reduce the sensitivity of the system to unknown perturbations. The following design ideas may be used.

1. Select the eigenvectors so that the spectral condition number (12), $\kappa(X)$, is minimized. This will give a rather good general insensitivity.

2. If we want to make some mode(s) as insensitive as possible, assign first the corresponding left and right eigenvectors to maximize the absolute values of the condition number(s) (1), s_i, for the corresponding eigenvalue(s). Then use the remaining design freedom to maximize other $|s_i|$:s in a proper order or in the way proposed in 1. If the eigenvalues are real, then this simply means that the eigenvectors should be selected (in a proper order) to be as orthogonal as possible.

3. If the perturbation is partly known, then it is possible to have better results. For example, suppose that the perturbation matrix has

an eigenvalue β and the corresponding eigenspace is $N(G-\beta I)$. Suppose that we may assign a closed loop eigenvector $\underline{x}_1$, corresponding to eigenvalue, say λ_1, so that $\underline{x}_1 \in N(G-\beta I)$. Then, since

$$\gamma_{i1} = y_i^T G x_1 = \beta\, y_i^T x_1 = \begin{cases} \beta \cdot s_1, & i = 1 \\ 0 & i \neq 1 \end{cases}$$

it is directly seen from eq. (10) that the eigenvalue λ_1 is shifted to $\lambda_1 + \beta \cdot \varepsilon$ and the rest of the perturbation does not affect the eigenvalue $\lambda_1 + \beta \cdot \varepsilon$ at all. This is especially important when $\beta = 0$, because then the perturbation has no affect on the eigenvalue λ_1.

An analogous result may be derived also by treating the left eigenvectors corresponding to λ_1, and by selecting the left eigenvector of the closed loop in analogous way.

Note that we do not suppose ε to be small in the previous result. The remaining design freedom (if any is left) could now be used to minimize the sensitivity of the remaining eigenvalues along the lines described in 1 and 2.

EXAMPLE

Consider the following controllable system

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Suppose, that all the elements of A are subject to small parameter variations and the following, possibly large, perturbation

$$G = \begin{bmatrix} g_1 & g_1 & -2g_1 \\ g_2 & g_2 & -2g_2 \\ 0 & 0 & 0 \end{bmatrix}$$

where g_1 and g_2 are perturbation parameters, may occur. Suppose further that we want to stabilize the system by state feedback and assign one eigenvalue, say $\lambda_1 = -0.5$, in as robust a way as possible. Suppose that the large perturbation is seldom present, and the main objective is to reduce the effect of small pertubations on λ_1, when there is no large perturbations present, and we also want to eliminate the effect of the large perturbation as well as possible.

A simple calculation gives

$$N_\lambda = \begin{bmatrix} \lambda-1 & 1 \\ 0 & 1 \\ \lambda & 1 \end{bmatrix}$$

It is easy to see that the following vectors form a basis of the nullspace, $N(G)$, of the perturbation, $[1, 1, 1]^T$, $[2, 0, 1]^T$. So we may select $x_1 = \frac{1}{\sqrt{3}}[1, 1, 1]^T$ to be the eigenvector corresponding to $\lambda_1 = -0.5$.

In order to reduce the sensitivity of λ_1, we should make the other eigenvectors orthogonal to x_1 and, if possible, one of them should be in the nullspace of the large perturbation. If we select especially $\lambda_2 = -1$, we may find $x_2 = \frac{1}{\sqrt{2}}[1, -1, 0]^T$, so that these objectives will be fulfilled.

The third eigenvalue is well separated, if we select e.g. $\lambda_3 = -4$. A suitable eigenvector will then be $x_3 = \frac{1}{\sqrt{14}}[+2, -3, +1]^T$, so that $x_1 \perp x_3$. The corresponding left eigenvectors and condition numbers for eigenvalues are:

$$\lambda_1 = -0.5, \quad y_1 = \frac{1}{\sqrt{3}}[1, 1, 1]^T, \quad s_1 = 1.0000$$

$$\lambda_2 = -1, \quad y_2 = \frac{1}{\sqrt{42}}[4, 1, -5]^T, \quad s_2 = 0.3273$$

$$\lambda_3 = -4, \quad y_3 = \frac{1}{\sqrt{6}}[1, 1, -2]^T, \quad s_3 = -0.3273.$$

So we see that the condition number for eigenvalue λ_1 is as large as possible and the other $|s_i|$:s are not very small. So we may suppose, that this closed loop system will behave well under perturbations assumed. The feedback matrix F will be (up to four decimals)

$$F = \begin{bmatrix} 1.1667 & 1.1667 & -3.8333 \\ -2.8333 & -4.8333 & 5.1667 \end{bmatrix}.$$

Suppose now that we would have made another selection of closed loop eigenvectors, without taking care of the condition numbers for eigenvalues. Another choise would be e.g. (up to four decimals)

$$\lambda_1 = -0.5 \qquad x_1^* = \begin{bmatrix} 0.5774 \\ 0.5774 \\ 0.5774 \end{bmatrix} \qquad y_1^* = \begin{bmatrix} 0.7053 \\ 0.0705 \\ -0.7053 \end{bmatrix}$$

$$s_1 = 0.0407$$

$$\lambda_2 = -1 \qquad x_2^* = \begin{bmatrix} 0.5111 \\ 0.6389 \\ 0.5750 \end{bmatrix} \qquad y_2^* = \begin{bmatrix} 0.6172 \\ 0.1543 \\ -0.7715 \end{bmatrix}$$

$$s_2 = 0.0296$$

$$\lambda_3 = -4 , \quad x_3^* = \begin{bmatrix} 0.3941 \\ 0.7881 \\ 0.4729 \end{bmatrix} , \quad y_3^* = \begin{bmatrix} 0.4082 \\ 0.4082 \\ -0.8165 \end{bmatrix} ,$$

$$s_3 = 0.0965.$$

Clearly x_1^*, $x_2^* \in N(G)$, but this selection of eigenvectors is more sensitive to small unknown perturbations, because the coefficients $|s_i|$ are smaller. The feedback matrix is now

$$F^* = \begin{bmatrix} -1.0000 & -5.5000 & 5.0000 \\ -5.0000 & -11.500 & 14.0000 \end{bmatrix}$$

Both the closed loop systems matrices were perturbed by small random numbers. A typical result is shown below. The perturbation matrix was

$$\Delta G = \begin{bmatrix} 0.0053 & 0.0005 & -0.0069 \\ -0.0067 & 0.0209 & -0.0039 \\ 0.0087 & 0.0089 & -0.0134 \end{bmatrix} .$$

The eigenvalues for different matrices are given in Table I.

TABLE I

A+BF+ΔG	-0.4954	-1.0025	-3.9892
A+BF*+ΔG	-0.5479	-0.9518	-3.9875
A+BF	-0.5000	-1.0000	-4.0000

The effect of perturbation on λ_1 (and on the other eigenvalues) is clearly smallest in the first case. In addition, the eigenvalues $\lambda = -0.5$ and $\lambda = -1$ are not affected by the large perturbation in either case

CONCLUSIONS

An algebraic design method to assign the closed loop eigenvalues and eigenvectors to guarantee internal robustness against parameter variations in the system matrix has been discussed. An example has been given to illustrate the ideas. Finally, we note that it is possible to apply the method to output feedback control by using the design freedom in eigenvector assignment found by Porter and Bradshaw [6]. On the other hand, by allowing completely free small variations in A, we ignore special features of plant structure. While this assumption is somewhat unrealistic, the results are not unduly restrictive from a practical point of view.

REFERENCES

[1] Wonham W.M., "Linear Multivariable Control, A Geometric Approach", Springer-Verlag, Berlin-Heidelberg-New York, 1974.

[2] Ramar K., Gourishankar V., "Utilization of the design freedom of pole assignment feedback controllers of unrestricted rank", Int. J. Control, 1976, Vol. 24, No 3.

[3] Shah S.L., Fisher D.G., Seborg D.E., "Eigenvalue invariance to system parameter variations by eigenvector assignment", Int. J. Control, 1977, Vol. 26, No 6.

[4] Wilkinson J.H., "The Algebraic Eigenvalue Problem", Clarendon Press. Oxford 1965.

[5] Moore B.C., "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment", IEEE Trans. Aut. Control, October 1976.

[6] Porter B., Bradshaw A., "Design of linear multivariable continuous - time output-feedback regulators", Int. J. Systems Sci., 1978, Vol. 9, No 4.

A NEW SPECTRAL SYNTHESIS PROCEDURE FOR MULTIVARIABLE REGULATORS

R. A. MAYNARD, R. R. MIELKE, S. R. LIBERTY AND S. SRINATHKUMAR
Department of Electrical Engineering
Old Dominion University
Norfolk, Virginia 23508

ABSTRACT

A method for selecting a multivariable state feedback controller that simultaneously achieves an a priori specification on closed loop eigenvalues and good mode mixing is presented. The problem is solved by projecting a desired modal matrix onto a constraint set containing the null space of the closed-loop state matrix, while assuring that the projection is in the null space. The feedback matrix follows immediately in the formulation. An example involving a helicopter hover controller is presented.

I. INTRODUCTION

This paper is concerned with the design problem of selecting the stationary feedback matrix K in the control law $u = Kx$ for a controllable multivariable system $\dot{x} = Ax + Bu$ where $x \varepsilon R^n$ and $u \varepsilon R^m$. The matrix K is to be selected so that the closed-loop system matrix A + BK has arbitrarily assigned eigenvalues. Once the eigenvalues are assigned, the remaining freedom available in selecting eigenvectors is utilized to achieve desirable modal behavior. Srinathkumar and Rhoten in [1], [2], and [3] have presented a "spectral synthesis algorithm" for constructing the feedback gain K which exploits the nonuniqueness of the modal matrix of A + BK in an attempt to achieve a priori design specifications. However, the a priori specifications are not explicitly incorporated in the procedure and the inherent design freedom available is not systematically exploited.

The work reported here relies heavily on insight gained in [1] but is based upon a different approach to the synthesis problem that allows a priori modal specifications, and results in a simple design procedure that can be parameterized and iterated.

II. NULL SPACE FORMULATION OF THE PROBLEM

The eigenvalue/eigenvector assignment problem can be expressed in the form $\hat{A}U = U\Lambda$, where $\hat{A} = A + BK$, U is the modal matrix and Λ is a diagonal matrix whose nonzero entries are the desired eigenvalues of $\hat{A}$.

This work was supported by the NASA-Langley Research Center under grant number NSG-1519.

This can be expressed in the form

$$[(I_n \otimes \hat{A}) + (-\Lambda^T \otimes I_n)] \text{ vec } U = 0, \tag{1}$$

where $\otimes$ is the Kronecker product [4] and vec $U = \text{col }(\underline{u}_i)$. Thus, if U is a solution to $\hat{A}U = U\Lambda$, then vec U lies in the null space of $S = [(I_n \otimes \hat{A}) + (-\Lambda^T \otimes I_n)]$. Moreover, $S \cdot \text{vec } U = 0$ if and only if $[\hat{A} - \lambda_i I_n]\, u_i = 0$ for all $i = 1, \cdots, n$.

In the remainder of this paper it is assumed that the system is in rank-reduced form with $B = [I_m \mid 0]^T$. If this is not the case, a coordinate transformation is applied to the system so that it is in this form. A solution is obtained according to the procedure to be described, and this solution is then transformed back to the original system coordinates. If A is partitioned so that

$$A = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]$$

with A_{11} an m x m matrix, then $[\hat{A} - \lambda_i I_n]\, u_i = 0$ if and only if the following two equations hold:

$$([A_{11} - \lambda_i I_m | A_{12}] + K)\, u_i = 0 \tag{2}$$

and

$$[A_{21} | A_{22} - \lambda_i I_{n-m}]\, u_i = 0 \tag{3}$$

Equation (3) is a constraint on the vectors that lie in the null space of $[\hat{A} - \lambda_i I_n]$; that is, vectors in the null space of $[\hat{A} - \lambda_i I_n]$ must also lie in the null space of $[A_{21} | A_{22} - \lambda_i I_{n-m}]$. Equation (2) is regarded as a constraint on K to be satisfied after U has been computed.

The desired modal mix is expressed as a desired modal matrix, P. By the projection theorem [5], the closest approximation, in a least squares sense, to P that can be achieved is obtained by projecting vec P onto the null space of S. This is equivalent to projecting p_i onto the null space of $[\hat{A} - \lambda_i I_n]$ for each $i = 1, \cdots, n$ to get each of the columns of the modal matrix, U, that most closely approximates P. This projection procedure can result in a U that is not a valid modal matrix. This occurrence will be indicated by the noninvertibility of U. When this happens, the entries of P are perturbed and the new P is projected. Once an invertible U is obtained, $\hat{A}$ is computed as $\hat{A} = U\Lambda U^{-1}$ with $\hat{A}$ partitioned to conform with A. The feedback matrix K is calculated from

$$K = [\hat{A}_{11}|\hat{A}_{12}] - [A_{11}|A_{12}] \tag{4}$$

Complex eigenvalues are treated in essentially the same way as real eigenvalues, after some preliminary manipulation. Using $\lambda_i = \alpha_i \pm j\beta_i$ and $u_i = x_i \pm jy_i$, $\hat{A}u_i = u_i\lambda_i$ becomes $\hat{A}(x_i \pm jy_i) = (x_i \pm jy_i)(\alpha_i \pm j\beta_i)$. The real and imaginary parts of this equation can be separated and the resulting equations can be written as

$$\hat{A}[x_i|y_i] = [x_i|y_i]\left[\begin{array}{c|c} \alpha_i & \beta_i \\ \hline -\beta_i & \alpha_i \end{array}\right] . \tag{5}$$

Partitioning

$$[x_i|y_i] = \left[\begin{array}{c|c} s_i & t_i \\ \hline v_i & w_i \end{array}\right] , \tag{6}$$

where s_i and t_i represent the first m-components of x_i and y_i, respectively, we obtain two constraint equations on u_i:

$$A_{21}\, s_i + A_{22}\, v_i = \alpha_i\, v_i - \beta_i\, w_i , \tag{7}$$

$$A_{21}\, t_i + A_{22}\, w_i = \beta_i\, v_i + \alpha_i\, w_i . \tag{8}$$

These can be expressed in the form

$$\left[\begin{array}{c|c|c|c} A_{21} & A_{22} - \alpha_i I_{n-m} & 0 & \beta_i I_{n-m} \\ \hline 0 & -\beta_i I_{n-m} & A_{21} & A_{22} - \alpha_i I_{n-m} \end{array}\right]\left[\begin{array}{c} x_i \\ \hline y_i \end{array}\right] = 0. \tag{9}$$

Thus, the constraint on u_i and its conjugate is that the 2-n component vector $(x_i^T\ y_i^T)^T$ must lie in the null space of

$$S_i = \left[\begin{array}{c|c|c|c} A_{21} & A_{22} - \alpha_i I_{n-m} & 0 & \beta_i I_{n-m} \\ \hline 0 & -\beta_i I_{n-m} & A_{21} & A_{22} - \alpha_i I_{n-m} \end{array}\right] \tag{10}$$

Letting p_i and q_i denote the real and imaginary parts, respectively, of the desired eigenvector conjugate pair associated with $\alpha_i \pm j\beta_i$, $(p_i^T\ q_i^T)^T$ is projected onto the null space of S_i to obtain the columns for the real and imaginary parts of the corresponding eigenvector conjugate pair in the system modal matrix.

III. HELICOPTER HOVER CONTROLLER APPLICATION

The design procedure is illustrated by an application to a problem considered by Srinathkumar [1]. The Sikorsky SH-3D helicopter in a hover mode is modeled as a ninth order linear system having state variables of longitudinal (u), lateral (v), and vertical (w) velocities in feet/seconds; pitch (q), roll (p) and yaw (r) rates in degrees/seconds; and pitch (Θ), roll (ϕ) and yaw (ψ) angles in degrees. The inputs are main rotor collective pitch (u_C), tail rotor collective pitch (u_T), longitudinal cyclic pitch (u_p) and lateral cyclic pitch (u_R), all in degrees.

The open-loop dynamics are given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \left[\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array}\right] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

where $x_1 = [u\ w\ q\ \Theta]^T$ and $u_1 = [u_p\ u_c]^T$ are the longitudinal variables and controls, and $x_2 = [v\ p\ \phi\ r\ \psi]^T$ and $u_2 = [u_R\ u_T]^T$ are the lateral variables and controls. The normalized system matrices, scaled by a rotor tip speed of 680 ft/sec, are given in Table I, along with the open-loop eigenvalues. The desired closed-loop eigenvalues are:

Response Variable	Eigenvalue
u	-4.5
w	-.324
q,Θ	-1.5 ± j1
v	-.3
p,ϕ	-1.5 ± j1
r,ψ	-1.5 ± j1

With mode decoupling as an objective, a block diagonal P matrix is specified. The resulting modal matrix U and control law matrix K are given in Table II. As a second example, another block diagonal P matrix and the resulting U and K matrices are given in Table III.

If the modal behavior resulting from the above procedure is unsatisfactory, then utilization of a weighted norm in the projection procedure allows the designer to iterate to a more satisfactory solution for the preselected P matrix. For example, if an element of the modal matrix resulting from a projection is too large, then the norm over which optimization is carried out (via the projection theorem) should be weighted more heavily in the corresponding component. This weighted norm idea, which is not exemplified here, allows one to carry out an iterated design procedure.

$$A = \begin{bmatrix} -0.0160 & 0 & 0.0297 & -0.5934 & -0.0047 & -0.0012 & 0 & 0 & 0 \\ 0 & -0.3242 & 0.0021 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1666 & 0.0843 & -0.5420 & 0 & 0.0620 & 0.5480 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0047 & -0.0007 & -0.0285 & 0 & -0.0330 & -0.0297 & 0.5934 & 0.0107 & 0 \\ 0.2199 & -0.0137 & -1.9400 & 0 & -0.6109 & -1.9600 & 0 & 0.0100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0.0013 & -0.0163 & -0.0083 & 0 & 0.4710 & -0.0043 & 0 & -0.3030 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$B^T = \begin{bmatrix} 60.7609 & -5.0318 & -2.13 & 0 & 0.1187 & 0.6900 & 0 & 0.005 & 0 \\ 0.5934 & 0 & -6.15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5934 & 21.8100 & 0 & 0.174 & 0 \\ 0 & 0 & 0 & 0 & 0.2611 & 0.3475 & 0 & -7.480 & 0 \end{bmatrix}$$

Open-loop Eigenvalues: $0, -0.305, -0.324, 0.08 \pm j0.313, -1.31 \pm j0.65, -0.047 \pm j0.414$

TABLE I. SYSTEM PARAMETERS

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} -0.7134 & -0.1770 & -0.4700 & 5.9545 & -0.0102 & -0.4556 & 0.3550 & 0.0492 & 0.0058 \\ 0.0637 & 1.0015 & 0.0640 & -0.5463 & 0.0022 & 0.0425 & -0.0252 & -0.0047 & -0.0011 \\ 0.0183 & 0.0000 & 0.5371 & -2.2397 & -0.0018 & 0.0148 & -0.0115 & -0.0019 & -0.0003 \\ -0.0041 & -0.0000 & -0.9370 & 0.8685 & 0.0060 & -0.0104 & 0.0007 & 0.0008 & 0.0007 \\ -0.0016 & 0.0006 & -0.0236 & 0.0312 & 0.7962 & 0.4001 & -0.1233 & -0.0142 & 0.0805 \\ -0.0030 & -0.0004 & -0.0077 & -0.0162 & 0.1216 & 0.4419 & -2.2530 & 0.0217 & -0.0107 \\ 0.0007 & 0.0011 & -0.0014 & 0.0098 & -0.4052 & -0.8972 & 0.9039 & -0.0133 & -0.0017 \\ -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0001 & -0.0006 & 0.5274 & -2.3079 \\ 0.0000 & -0.0000 & 0.0000 & -0.0000 & 0.0000 & -0.0002 & 0.0002 & -0.9536 & 0.9029 \end{bmatrix}$$

$$K = \begin{bmatrix} -0.0776 & -0.0015 & -0.1145 & 0.0722 & 0.0197 & 0.0121 & 0.0459 & -0.0013 & -0.0037 \\ 0.0439 & 0.0140 & 0.4076 & 0.4903 & 0.0024 & 0.0854 & -0.0113 & 0.0001 & 0.0006 \\ -0.0079 & 0.0009 & 0.0912 & -0.0009 & -0.0487 & -0.0596 & -0.1895 & -0.0087 & -0.0068 \\ -0.0001 & -0.0022 & 0.0010 & 0.0000 & 0.0618 & -0.0020 & -0.0044 & 0.3604 & 0.4343 \end{bmatrix}$$

TABLE II. FIRST ITERATION

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.5 & -0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & -0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.5 & -0.5 \end{bmatrix}$$

$$U = \begin{bmatrix} -0.7134 & -0.2650 & -0.4180 & -0.6064 & -0.0102 & -0.3018 & -0.1650 & 0.0459 & -0.0283 \\ 0.0637 & 15.0220 & 0.0195 & 0.0367 & 0.0022 & 0.0253 & 0.0165 & -0.0048 & 0.0028 \\ 0.0183 & 0.0001 & 1.2695 & 0.2324 & -0.0018 & 0.0076 & 0.0056 & -0.0017 & 0.0011 \\ -0.0041 & -0.0002 & -0.5144 & -0.4979 & 0.0060 & -0.0018 & -0.0050 & 0.0011 & 0.0000 \\ -0.0016 & 0.0088 & -0.0118 & -0.0124 & 0.7962 & 0.0581 & 0.2039 & -0.0379 & -0.0088 \\ -0.0030 & -0.0055 & 0.0078 & -0.0037 & 0.1216 & 1.1359 & 0.2463 & 0.0088 & -0.0239 \\ 0.0007 & 0.0170 & -0.0048 & -0.0007 & -0.4052 & -0.4485 & -0.4632 & -0.0114 & 0.0083 \\ -0.0000 & 0.0000 & -0.0000 & -0.0000 & -0.0000 & 0.0000 & 0.0001 & 1.2637 & 0.2363 \\ 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & -0.5105 & -0.4979 \end{bmatrix}$$

$$K = \begin{bmatrix} -0.0742 & -0.0014 & 0.0223 & 0.0905 & 0.0167 & 0.0020 & 0.0366 & 0.0044 & 0.0036 \\ 0.0435 & 0.0140 & 0.3939 & 0.5094 & 0.0020 & 0.0865 & -0.0115 & -0.0008 & -0.0002 \\ -0.0080 & 0.0009 & 0.0883 & -0.0005 & -0.0486 & -0.0594 & -0.1893 & -0.0088 & -0.0070 \\ -0.0001 & -0.0022 & 0.0010 & 0.0000 & 0.0618 & -0.0020 & -0.0044 & 0.3604 & 0.4343 \end{bmatrix}$$

TABLE III. SECOND ITERATION

REFERENCES

[1] S. Srinathkumar, "Spectral Characterization of Multi-Input Dynamic systems", Ph.D. dissertation, Oklahoma State University, Stillwater, Oklahoma, May 1976.

[2] S. Srinathkumar, "Eigenvalue/Eigenvector Assignment Using Output Feedback", IEEE Trans. Automatic Control, Vol. AC-23, pp. 79-81, February 1978.

[3] S. Srinathkumar and R. P. Rhoten, "Linear Multivariable Synthesis By Eigenvalue/Eigenvector Assignment", Proc., International Forum on Alternatives for Multivariable Control, pp. 164-174, October 1977.

[4] R. Bellman: Introduction To Matrix Analysis, McGraw-Hill Book Company, Inc., New York, 1970.

[5] D. G. Luenberger: Optimization By Vector Space Methods, John Wiley and Sons, Inc., New York, 1969.

DECENTRALIZED STABILIZATION OF LARGE-SCALE SYSTEMS USING MODAL CONTROL AND EIGENVALUE SENSITIVITIES

Z. ELRAZAZ and N.K. SINHA
Group on Simulation, Optimization and Control
McMaster University
Hamilton, Ontario, Canada L8S 4L7

ABSTRACT

Modal control and eigenvalue sensitivities together with multi-stage control have been used to construct a new algorithm for designing local and global state feedback for decentralized stabilization of large-scale systems. This approach can handle the constraints on the gain vectors as well as the direction of information flow between the subsystems.

INTRODUCTION

The problem of stabilizing large-scale systems through decentralized feedback has been a subject of considerable interest in recent years as appears from the comprehensive survey by N.R. Sandell et al. [1]. These authors have considered linear time-invariant continuous systems modelled by the following equations:

$$x(t) = Ax(t) + \sum_{i=1}^{m} B_i u_i(t) \tag{1}$$

$$y_i(t) = C_i x(t) \tag{2}$$

where $x \in R^n$, $u_i \in R^{p_i}$, and $y_i \in R^{q_i}$, $i = 1, 2, \ldots, m$ are, respectively, the states of the system, the input, and the output of the i^{th} controller. Wang and Davison [2], and Corfmat and Morse [3] have considered stabilizability of such systems by applying the following decentralized output feedback control

$$u_i(t) = M_i z_i(t) + F_i y_i(t) + G_i v_i(t) \tag{3}$$

$$z_i(t) = H_i z_i(t) + L_i y_i(t) + R_i v_i(t) \tag{4}$$

where $z_i(t)$ is the state of the dynamic compensator of chosen order and $v_i(t)$ is an external signal of appropriate dimension. The matrices M_i, F_i etc. are available to the designer for placing the eigenvalues of the augmented system (obtained by adjoining equations (1) and (4)) on the open left-half of the complex plane. They derived necessary condition for stabilizing the system using constant output feedback and it turns out to be necessary and sufficient for dynamic output feedback. The condition is that the fixed modes [2] of the system have strictly negative real parts. Their method for stabilization is to use static feedback (implicit in Davison's algorithm) to make the system controllable and observable from one or more controllers and then design a dynamic compensator for this controller to stabilize the whole augmented system. Siljak and Vukcevic [4, 5] have also studied the question of stabilizing large-scale systems through decentralized constant state feedback. These authors have considered the control law in the following hierarchical form:

$$u_i(t) = - K_i^T x_i(t) - \sum_{\substack{j=1 \\ j\neq i}}^{m} K_{ij}^T x_j(t) \tag{5}$$

where $x_i(t)$ is the n_i-dimensional state of the i^{th} subsystem, K_i and K_{ij} are the gain vectors representing the feedback of the local and global states. By making use of Siljak's Lyapunov thoeretic condition for stability of large scale systems, these authors have developed a technique for computing the gain vectors K_i and K_{ij}. Their procedure is an iterative one and if Sevastyaniv-Kotelyanski conditions are not satisfied at the last step, one has to go to the first step again. They have included examples for which the gains are rather large and impractical. But it should be clear that their approach is quite good because most of the computations are carried out on the subsystem level.

In this paper a new approach is described for stabilizing the system using decentralized feedback control by applying local feedback and small part (if necessary) of global feedback. This approach is based on the ideas of partial modal control, multi-stage control, and eigenvalue sensitivities. To improve the transient response characteristics, a local integral controller is implemented. The feedback gains obtained are more reasonable than those obtained by Siljak [5]. In addition, there is no transformation required which helps in preserving the physical interpretation of the states. Moreover, constraints on the gain vectors and on direction of information flow between subsystems can be taken into consideration.

Consider the linear time invariant continuous system S described by the equations

$$x_i(t) = A_i x_i(t) + \sum_{\substack{j=1 \\ j\neq i}}^{s} A_{ij} x_j(t) + B_i u_i(t) \tag{6}$$

$$y_i(t) = C_i\, x_i(t) \tag{7}$$

where $x_i \in R^{ni}$, $u_i \in R^{mi}$, and $y_i \in R^{qi}$ are the states, the input and the output of the i^{th} subsystem, respectively.

DEFINITION

The system S is said to be stabilizable with decentralized feedback (5) if there exist a set of $m_i \times n_i$ constant matrices K_i (and $K_{ij} \in \mathbf{R}^{m_j x n_j}$ if global control is necessary), i = 1, 2, ..., s such that the eigenvalues of the closed loop system lie in the open left-half of the complex plane.

NOTE: The fixed modes of the system should have strictly negative real parts.

PROPOSITION

The system S can be stabilized using partial modal feedback control and eigenvalue sensitivity analysis.

PARTIAL MODAL FEEDBACK CONTROL

Let $\lambda_1^{(k)}, \lambda_2^{(k)}, \ldots, \lambda_n^{(k)}$ in descending order (with respect to their real parts) and $V_1^{(k)}, V_2^{(k)}, \ldots, V_n^{(k)}$ be the eigenvalues and the eigenvectors of the transpose of the closed loop matrix at stage k. So the feedback law at that stage is described by the equation

$$u_i^{(k)}(t) = -\, g_i^{(k)} x(t) \tag{8}$$

where

$$g_i^{(k)} = \sum_{j=1}^{n} \frac{\prod_{\ell=1}^{n} (\rho_\ell^{(k)} - \lambda_j^{(k)})}{v_j^{(k)^T} b_i \prod_{\substack{\ell=1 \\ \ell \neq j}}^{n} (\lambda_\ell^{(k)} - \lambda_j^{(k)})} V_j^{*(k)^T}$$

Here the elements of $v_j^{*(k)}$ are chosen to be the elements of $v_j^{(k)}$ which correspond to subsystem i and zero elsewhere. Also $\rho_\ell(k)$ are the eigenvalues being adjusted at the k^{th} iteration, whereas $\lambda_j(k)$ are all eigenvalues at the k^{th} iteration. $V_j^{(k)} b_i$ is the largest element in the j^{th} row of the mode-controllability matrix [6].

DEFINITION (Effective Eigenvalue Sensitivity Region)

It is the region in which one can use the eigenvalue sensitivities to assign the desired pole locations.

ALGORITHM

The algorithm is divided into two phases:

Phase I (Local Controller)

1. Eigen analysis (eigenvalues, eigenvectors)
    - Check for the required degree of exponential stability M (M = min Real $(-\lambda_i)$, i = 1, 2, ..., n), if all the eigenvalues have negative real parts. If it is satisfied, stop.
    - Check for the possibility of using eigenvalue sensitivities with respect to gain matrix elements to assign the required M. If it is possible, go to step 7. Otherwise, continue and check for the number of iterations. If it exceeds a specified number, go to Phase II.
2. Arrange the eigenvalues in descending order with respect to their real parts and arrange their corresponding eigenvectors in the same order.
3. Obtain rows of the mode-controllability [6] matrix ($P = v^T B$) which correspond to the undesired poles. This is easily done since $(V^T)^{-1} A\ v^T$ is in the diagonal form.
4. Test which input $u_i(t)$ has the largest effect on one or more undesired pole locations without affecting the stable poles severely.
5. Assume suitable locations $\rho_\ell^{(k)}$ for the undesirable poles locations (complex poles should be in pairs). Obtain the corresponding partial modal feedback law (8).
6. Obtain the new closed-loop matrix

$$\bar{A}^{(k)} = \bar{A}^{(k-1)} - b_i\ g_i^{(k)}, \quad \bar{A}^{(0)} = A \tag{9}$$

    GO TO 1

7. Using the eigenvalue sensitivities [6] assign the required pole locations.

    GO TO 1

Phase II

Using the eigenvalue sensitivites [6] with respect to the interconnection gains (global control), assign the required pole locations, taking into consideration the various constraints on the gain matrix elements and the direction of information.

Example (1) (Phase I)

Consider the system described by the equation

$$\dot{x}(t) = A\ x(t) + B\ U(t)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 3 & 4 \\ -2 & -1 & -1 & 2 & 1 \\ 4 & 0 & 0 & 0 & 1 \\ 5 & 6 & 0 & -3 & -2 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The open loop poles are:

$$5.1,\ 1.7,\ -1.3,\ -4.3 \pm j1.8$$

Using the suggested algorithm, the closed loop poles are:

$$-.9,\ -1.3 \pm .8j,\ -4.5,\ -25.$$

with

$$K_1^T = [37.6 \quad 25.5 \quad 6.],\ \text{and}\ K_2^T = [39.4 \quad 23.9]$$

There is still possibility for futher improvement if the degree of the exponential stabiliy M is not satisfied. The same example was considered by Siljak and Vukcevic [5]. Their procedure gave the following closed loop poles:

$$-25.9 \pm j3.5,\ -36.,\ -68. \pm j6.$$

with

$$K_1^T = [93748. \quad 6874. \quad 149.],\ \text{and}\ K_2^T = [1247 \quad 73.]$$

Example (2) (Phase II)

Consider the system described by the Pair [A, B] where

$$A = \begin{bmatrix} 1.00 & 11.50 & 86.50 & 4.00 & 22.50 \\ .45 & 0.00 & -4.09 & 8.91 & -.82 \\ 0.18 & 1.00 & 8.36 & .36 & 3.27 \\ 0.00 & 1.25 & 14.75 & 5.00 & 2.75 \\ 0.18 & 0.00 & -9.82 & .18 & -6.36 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

The open loop poles are:

$$11.54,\ .76 \pm j1.83,\ -1.15,\ -3.89$$

Using the suggested algorithm the closed loop poles are:

$$-.54 \pm j1.9, -.64, -.8, -18.9$$

with

$$K_1^T = [2.07 \quad 10.23 \quad 72.75], \; K_{12}^T = [3. \quad 10.6]$$

$$K_{21}^T = [0. \quad 0. \quad 0.], \text{ and } K_2^T = [7.46 \quad 9.67]$$

NOTE: No information flows from subsystem (1) to subsystem (2).

The same example was also considered by Siljak and Vukcevic [5] and the closed loop poles were found to be -1. ± j.16, -10.3, -11.9, -15.2 with

$$K_1^T = [-127.58 \quad 96.33 \quad 105.29] \; K_{12}^T = [166.25 \quad 36.63]$$

$$K_{21}^T = [.05 \quad .68 \quad .62], \text{ and } K_2^T = [.2 \quad -.7]$$

DECENTRALIZED INTEGRAL LOCAL CONTROL

In order to design decentralized control system which possesses good steady-state as well as good transient response characteristics, it is necessary to introduce decentralized integral as well as proportional feedback of some of the subsystem state variables.

Consider the equation (6) which describes the system. If integrals of ℓ ($\ll n$) of the elements of x(t) which correspond to the subsystem states are also to be fedback, the state vector of the resulting system will consist of x(t) augmented by the $\ell \times 1$ vector $\chi(t)$ where

$$\dot{\chi}(t) = E_\ell \, x(t) \tag{10}$$

and E_ℓ is an $\ell \times n$ matrix which consists of an appropriate selection of ℓ rows from the $n \times n$ unit matrix I_n. The state equation of the augmented system is then

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\chi}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ E_\ell & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \chi(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} U(t) \tag{11}$$

NOTE: The choice of E_ℓ should not destroy the controllability of a priori controllable systems.

Example (3)

Consider the system given in example (1) and choose E_ℓ to be

$$E_\ell = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The response of the resulted augmented system after introducing the integral control is compared with the previous response in Fig. 1.

CONCLUSION

Partial modal feedback control and eigenvalue sensitivities have been used to design local and global controllers for stabilizing decentralized control systems. This approach can handle the constraints imposed on the system structure and on the gain vectors. There is no transformation of the original states required, which helps in preserving the physical interpretation of the states. This approach can be used to shape the output responses by assigning partial eigenvectors components.

REFERENCES

[1] N.R. Sandell, et al. "Survey of decentralized control methods for large scale systems", IEEE Trans. on Auto. Cont., AC-23, pp. 108-129, 1978.

[2] S.H. Wang, and E.J. Davison, "On the stabilization of decentralized control systems", IEEE Trans. on Auto. Cont., AC-18, pp. 473-478, 1973.

[3] R.P. Corfmat and A.S. Morse, "Stabilization with decentralized feedback control", IEEE Trans. Auto. Cont., AC-18, pp. 679-682, 1973.

[4] D.D. Siljak, and M.B. Vukcevic, "Large-scale systems: Stability, complexity, reliability", Journal of the Franklin Institute, vol. 301, nos. 1 and 2, Jan.-Feb. 1976, pp. 49-69.

[5] D.D. Siljak, and M.B. Vukcevic, "Decentralization, stabilization and estimation of large-scale linear systems", IFAC Symposium on Large-Scale Systems Theory and Applications, Udine, Italy, June 16-22, 1976.

[6] B. Porter and R. Crossley. Modal Control, Theory and Applications. London: Taylor and Francis, 1972.

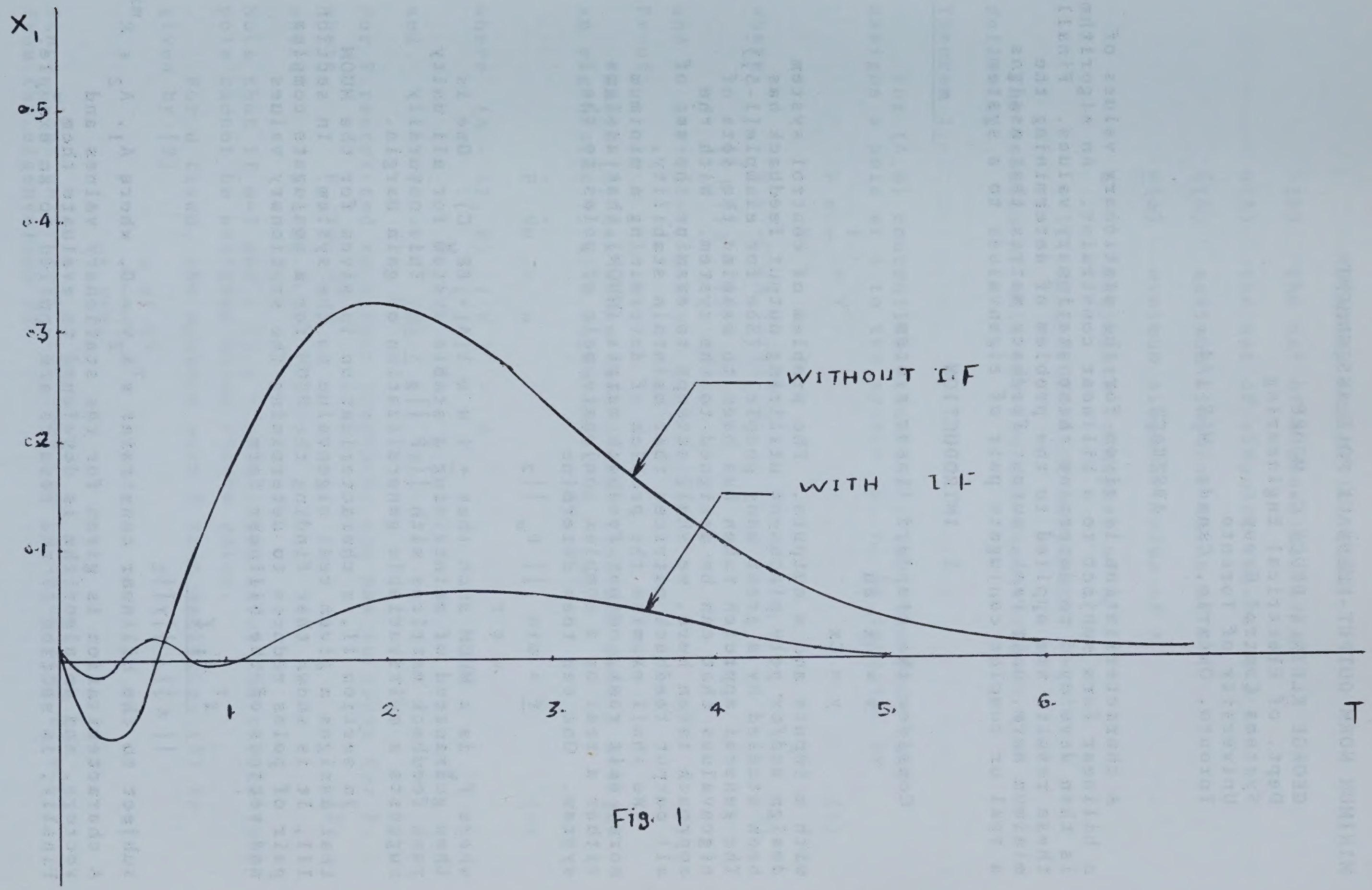


Fig. 1

MINIMUM NORM OUTPUT-FEEDBACK POLE ASSIGNMENT

GEORGE KLEIN & BRUCE C. MOORE
Dept. of Electrical Engineering
Systems Control Group
University of Toronto
Toronto, Ontario, Canada M5S 1A4

ABSTRACT

A characterization is given for the stationary values of a bilinear form subject to a bilinear constraint. An algorithm is then developed to determine these stationary values. Finally, these results are applied to the problem of determining the minimum norm, unit rank, output feedback matrix that assigns a real or complex conjugate pair of eigenvalues to a system.

I. INTRODUCTION

Consider the standard linear system

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

with m inputs and m outputs. The problem of control system design and/or pole placement utilizing output feedback has been studied by a great many people. (See for example[1-5]). The general approach taken has been to examine the sets of eigenvalues that can be assigned to the system. With the approach taken here, we shall attempt to examine the set of all output feedback matrices that maintain stability.

We shall examine the problem of determining a minimum norm, unit rank, output feedback matrix (MUOM) that assigns either a real or a complex conjugate pair of poles to the system. One can then determine

$$\underline{f} = \min_{w \in r} \| F_w \|_2$$

where F_w is a MUOM such that $\pm i w \in \lambda(A + BF_w C)$. One is then guaranteed of maintaining a stable system for all unity rank feedback matrices with $\| F \|_2 \le \underline{f}$. This naturally suggests a multivariable generalization of gain margin.

In section II, a characterization is given for the MUOM that assigns a given real eigenvalue to the system. In section III, it is shown that finding the MUOM for a conjugate complex pair of poles reduces to determining the stationary values and vectors of the bilinear form

$$\frac{x^T A_1 y}{\| x \|_2 \, \| y \|_2}$$

subject to the bilinear constraint $x^T A_2 y = 0$, where $A_1, A_2 \in R^{m \times m}$. A characterization is given for the stationary values and vectors, and an algorithm is developed to evaluate them. Finally, in section IV the results are applied to an example.

NOTATION AND DEFINITIONS

$\lambda(A)$	the set of eigenvalues of A.
$\sigma(A)$	the set of singular values of A
$\bar{\sigma}(A)$	maximum singular value of A
$\underline{\sigma}(a)$	minimum singular value of A
A^*	conjugate transpose of A

II. REAL EIGENVALUE ASSIGNMENT

The results of this section are summarized by the following:

Theorem 1.

For (A,B) controllable and $\lambda \notin \lambda(A)$, the MUOM that assigns a pole at λ to the system (C, A, B) is given by

$$F = \frac{1}{\gamma} u v' \qquad (1)$$

where $$V = -C(A - \lambda I)^{-1} B \qquad (2)$$

$$||u||_2 = ||v||_2 = 1$$

and $$u^*V^*Vu = v^*VV^*v = \gamma^2 = \bar{\sigma}(V) \qquad (3)$$

Proof:

Moore [6] has shown that a real feedback matrix assigns an eigenvalue γ, with eigenvector $\hat{V}u$ iff

$$\hat{F}\hat{V}u = u \qquad (4)$$

where $$(A - \lambda I, B)\begin{pmatrix} \hat{V} \\ I \end{pmatrix} = 0$$

and $$\hat{V} = -(A - \lambda I)^{-1} B$$

For $\hat{F}$ restricted to output feedback, one has to solve for F,

where $$F\,Vu = u \qquad (5)$$

where $$V = C\hat{V}$$

Note that if m=1 and λ is a transmission zero, the closed loop pole cannot be assigned using finite gains.

For u fixed, the minimum norm F that satisfies (5) is given by [9]

$$F = \frac{u\,u^*\,V^*}{u^*\,V^*Vu} \qquad (6)$$

Since F has unity rank, its norm can be calculated directly from the eigenvalues of

$$FF^* = \frac{uu^*}{u^* V^* Vu}$$

Since the nonzero eigenvalues of XY are the same as those of YX [7]

$$\min_{||u||=1} ||F||_2^2 = \frac{1}{\max\limits_{||u||=1} u^*V^*Vu} = \frac{1}{\bar{\sigma}^2(V)} \quad (7)$$

For u, the normalized right singular vector corresponding to $\sigma(V)$, one has [7]

$$\sigma v = Vu$$

Substitution of these results into (6) proves the theorem ☐

III. COMPLEX EIGENVALLUE ASSIGNMENT

As in section II, it can be shown that the output feedback matrix that assigns poles at $\pm$ i w satisfies the following equations:

$$[A - i w I, B] \begin{vmatrix} \hat{V}_1 + i \hat{V}_2 \\ I \end{vmatrix} = 0 \quad (8)$$

$$\hat{V}_1 = A (A^2 + W^2 I)^{-1} B \quad (9)$$

$$\hat{V}_2 = W (A^2 + W^2 I)^{-1} B \quad (10)$$

$$V_j = C \hat{V}_j \quad j = 1, 2 \quad (11)$$

$$F(V_1 k_{.} - V_2 k_2, V_1 k_2 + V_2 k_1) = (k_1, k_2) \quad (12)$$

The last equation is a result of restricting F to be real and examining the real and imaginary parts of

$$k = k_1 + i k_2 \quad (13)$$

$$F (V_1 + i V_2) k = k \quad (14)$$

Note that if an arbitrary complex conjugate pair of poles are to be assigned at (λ, λ^*), the following substitutions have to be made

$$A \to A - \lambda_r I, \quad \lambda_r = \text{Real} (\lambda)$$

$$w \to \text{Imag} (\lambda)$$

If F is restricted to have unity rank, then (12) implies that $k_1 = a\, k_2$ for some real a, and there exists θ such that $k\, e^{i\theta}$ is real. These results are summarized in the following:

Lemma 1.

The unity rank, output feedback matrix F, that assigns system poles at $\pm\, i\, w$, satisfies

$$F\ (V_1\, k,\quad V_2\, k) \;=\; k \tag{15}$$

$$F = \gamma\, k\, h^T \tag{16}$$

and

$$\|\, k \,\|_2 = \|\, h \,\|_2\,, \quad k,\, h\ \varepsilon\ R^m$$

$$\gamma \neq 0,\ \gamma\ \varepsilon\ R$$

Proof:

The first part was proved in the previous discussion, while the second part is an easy consequence of noting that F has unity rank and that Im F = k, □

The problem of finding the MUOM that assigns poles at $\pm\, i\, w$ then reduces to finding

$$\max_{\|h\|_2=\|k\|_2=1} \left[\frac{1}{\gamma}\right]$$

subject to

$$h^T V_1 k = \frac{1}{\gamma}$$

$$h^T V_2 k = 0$$

As with similar problems examined by Golub [8], this one can be approached using Lagrange multipliers to examine the stationary points of

$$g = h^* V_1 k + \lambda h^* V_2\, k + \frac{a}{2}\,(h^* h - 1) + \frac{b}{2}\,(k^* k - 1) \tag{17}$$

On setting the partial derivatives to zero, one has

$$g_h = g_k = 0$$

$$\Longrightarrow\quad D\ (\lambda,\ a,\ b)\ \begin{pmatrix} k \\ h \end{pmatrix} = 0 \tag{18}$$

where

$$D(\lambda, a, b) = \begin{bmatrix} V_1 + \lambda\, V_2 & aI \\ bI & (V_1 + V_2)^T \end{bmatrix} \tag{19}$$

Using elementary properties of determinants [7], one has

$$| D | = | (V_1 + \lambda V_2)(V_1 + \lambda V_2)^* - abI | = 0 \tag{20}$$

which implies $(a\ b)^{1/2} \ \varepsilon\ \sigma\ (V_1 + \lambda\ V_2)$

By symmetry one has

$$a = b\ \varepsilon\ \sigma\ (V_1 + \lambda\ V_2)$$

and D can be expressed as a function of the 2 variables λ and b. To identify b, note that

$$(V_1 + \lambda\ V_2)\ k = -\ b\ h$$

and

$$-b = h^T (V_1 + \lambda\ V_2)\ k = h^T V_1\ k = \frac{1}{\gamma} \tag{21}$$

Another property of D (λ, b) is

Lemma 2.

For $| b | \geq || V_1 ||_2$, the set of λ that satisfy $| D (\lambda, b) | = 0$ are all real.

Proof:

As in (20)

$$| D | = | \lambda^2 V_2 V_2^* + \lambda (V_2 V_1^* + V_1 V_2^*) - Q | = | P(\lambda) |$$

where $Q = b^2 I - V_1 V_1^* \geq 0$

For a λ such that $| P(\lambda) | = 0$, $\exists x$ such that

$$x^* P(\lambda) x = \lambda^2 a + \lambda b - c = 0 \tag{22}$$

where $a, c \geq 0$

The discriminant of (22) is $b^2 + 4\ a\ c \geq 0$

$\Rightarrow \lambda$ is real

Note that for a symmetric matrix S,

$x^* Sx$ is real even if x is complex: □

An easy extention of lemma 2 is

Lemma 3.

If $| b | > || V_1 ||_2$ then $| D (0,b) | \neq 0$

Proof:

Assuming $\lambda = 0$ contradiction easily follows by noting that Q, in lemma 2, is now strictly positive definite □

To further characterize the stationary values of g we have the following:

Lemma 4.

Assume V_2 is nonsingular. If $D/\lambda,b)\begin{pmatrix}k\\h\end{pmatrix} = 0$ and $h^T V_2\, k = 0$, then $-\lambda$ is a multiple eigenvalue of

$$G\ (b) = \left|\begin{matrix} V_2^{-1} & 0 \\ 0 & V_2^{*-1} \end{matrix}\right| D(0,b) \tag{23}$$

Proof:

With $h^* V_2\, k = 0$, we have

$$(h^*, k^*) \begin{pmatrix} V_2 & 0 \\ 0 & V_2^* \end{pmatrix} \begin{pmatrix}k\\h\end{pmatrix} = 0 \tag{24}$$

$$\Rightarrow \begin{pmatrix} V_2\, k \\ V_2^*\, h \end{pmatrix} \varepsilon \ \mathrm{Ker}\ (h^*, k^*) \tag{25}$$

Since $D\ (\lambda, b) \begin{pmatrix}k\\h\end{pmatrix} = 0$ one also has

$$(h^*\ k^*)\ D\ (\lambda, b) = 0$$

Case 1: If $Rk(D\ (\lambda,b)) = 2\ m - 1$

then $\begin{pmatrix}k\\h\end{pmatrix} = (G\ (b) + \lambda I) \begin{pmatrix}x\\y\end{pmatrix}$

for some $\begin{pmatrix}x\\y\end{pmatrix}$. Since $\begin{pmatrix}k\\h\end{pmatrix}$ is in the kernel as well as the image of $(G\ (b) + \lambda I)$, $-\lambda$ must be a multiple eigenvalue of $G\ (b)$ with a Jordan block of dimension at least 2.

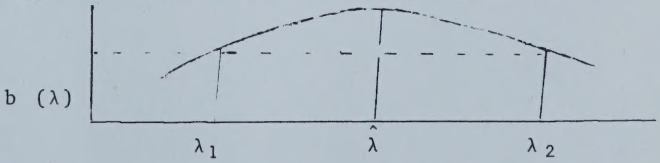
Case 2: If $Rk(D\ (\lambda,b)) = 2\ m - 2$

then $G(b)$ has an eigenvalue at $-\lambda$ with at least 2 grade 1 eigenvectors. □

Thus the stationary values of g are characterized by $G\ (b)$ having a double real root at $-\lambda$ with $g\ (\lambda,b) = b\ \varepsilon\ \sigma\ (V_1 + \lambda V_2)$ and thus

$$\frac{\partial b}{\partial \lambda} = 0$$

Because b has a stationary point with respect to λ, the following graph suggests an algorithm for computing both b and λ



For $| D (\lambda_j, b) | = 0, j \in \underline{2m}$, we have from (20),

$$b \in \sigma (V_1 + \lambda_j V_2), j \in \underline{2m}$$

Because b has a stationary value with respect to λ, we also have

$$b \ (\frac{\lambda_1 + \lambda_2}{2} \geq b (\lambda_1) = b (\lambda_2)$$

This results in the following algorithm

1) For $b_j < \| V_1 \|_2$ find $\lambda_{1_j}, \lambda_{2_j}$

2) For $\lambda_{3_j} = \frac{\lambda_{1_j} + \lambda_{2_j}}{2}$ find $b_{j+1} \in \sigma(V_1 + \lambda_{3j} V_2)$

3) Increase j and repeat

The results of this section are summarized in the following:

Theorem 2.

The MUOM F, that assigns system poles at ± i w, is characterized by

$$F = \gamma k h^T \qquad (26)$$

$$\| k \|_2 = \| k \|_2 = |, \quad k, h \in R^m \qquad (27)$$

$$(V_1 + \lambda V_2) k = \frac{1}{\gamma} h \qquad (28)$$

$$\frac{1}{\gamma} \in \sigma (V_1 + \lambda V_2)$$

and λ is a multiple root of $| D (\lambda, - \frac{1}{\gamma}) | = 0$ □

IV EXAMPLE: In the following example the system was chosen as:

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -0.02 \\ -0.8398 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -2.381 & 55.98 \\ 0 & 50 \end{bmatrix}$$

$$\lambda \quad (A) = \{ -1, \ -2 \}$$

The MUOM were calculated using the algorithm and the results are shown in fig 1. Note that the system is relatively close to instability as is seen from the resonance effect at $W \simeq 1.4$. That is, while the gain can generally be as high as .5, at the resonance frequency the gain margin drops to approximately .03.

CONCLUSIONS

A characterization was given for the minimum norm, unity mark, output feedback matrix that assigns a given real or complex conjugate pair of eigenvalues to a system. A characterization and an algorithm were also given for the stationary values of a bilinear form subject to a bilinear constraint.

REFERENCES

1. H. Kimura, A Further Result on the Problem of Pole Assignment by Output Feedback, IEEE Trans.A.C., Vol. 22, pp. 458-463, June 1977.

2. H. Kimura, Pole Assignment by Gain Output Feedback, I.E.E.E. Trans. A.C., Vol. 20, pp. 509-516, Aug. 1975.

3. E.J. Davison, S.H. Wang, On Pole Assignment in Linear Multivariable.

4. R.V. Patel, On Output Feedback Pole Assignability, Int. Journal of Control, Vol. 20, pp. 955-960, Dec. 1974.

5. R. Herman, C.F. Martin, Applications of Algebraic Geometry to Systems Theory - Part I, IEEE Trans. A.C. Vol. 22, pp. 19-25, Feb. 1977.

6. B.C. Moore, On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment, IEEE Trans . A.C., Vol 21, pp 689-691, Oct 1976.

7. B. Noble, Applied Linear Algebra, Crentice Mall, New Jersey, 1969.

8. G.H. Golub, Some Modified Matrix Eigenvalue Problems, Siam Rev., Vol. 15, No. 2, pp. 318-334, April 1973.

9. T.N.E. Greville, Adi Ben-Israel, Generalized Inverses - Theory and Applications, Wiley, New York, 1974

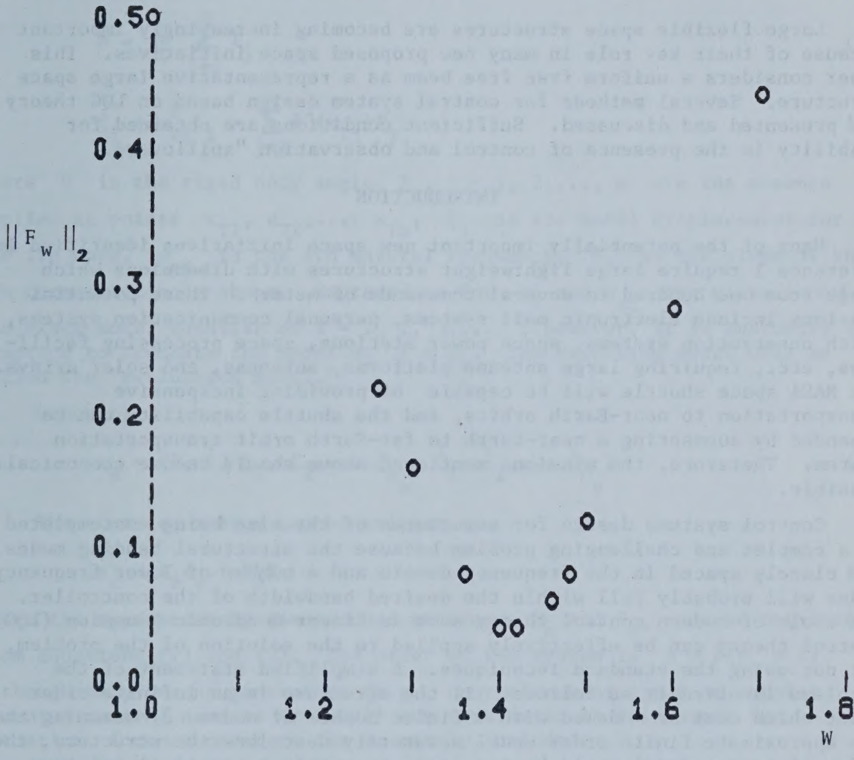


Figure 1

DESIGN OF REDUCED-ORDER CONTROLLERS FOR LARGE FLEXIBLE SPACE STRUCTURES

by

SURESH M. JOSHI
O.D.U. Research Foundation
Norfolk, VA 23508

NELSON J. GROOM
NASA Langley Research Center
Hampton, VA 23665

ABSTRACT

Large flexible space structures are becoming increasingly important because of their key role in many new proposed space initiatives. This paper considers a uniform free free beam as a representative large space structure. Several methods for control system design based on LQG theory are presented and discussed. Sufficient conditions are obtained for stability in the presence of control and observation "spillovers."

INTRODUCTION

Many of the potentially important new space initiatives identified in reference 1 require large lightweight structures with dimensions which range from one hundred to several thousands of meters. These potential missions include electronic mail systems, personal communication systems, Earth observation systems, space power stations, space processing facilities, etc., requiring large antenna platforms, antennas, and solar arrays. The NASA space shuttle will be capable of providing inexpensive transportation to near-Earth orbits, and the shuttle capability can be expanded by augmenting a near-Earth to far-Earth orbit transportation system. Therefore, the missions mentioned above should become economically feasible.

Control systems design for structures of the size being contemplated is a complex and challenging problem because the structural bending modes are closely spaced in the frequency domain and a number of lower frequency modes will probably fall within the desired bandwidth of the controller. The tools of modern control theory such as Linear-Quadratic-Gaussion (LQG) control theory can be effectively applied to the solution of the problem, but not using the standard techniques. A simplified statement of the problems involved is as follows: 1) the structure is an infinite order plant which must be modeled with a finite number of modes, 2) assuming that the approximate finite order model accurately describes the structure, the order of such a model would be too large to permit a practical solution using standard LQG theory with full order estimators and regulators, 3) the feasible solution, therefore, is almost certain to consist of regulators and estimators of order much lower than the plant.

Unfortunately, stability of the system with lower order regulators and estimators is not guaranteed because of the possible excitation of uncontrolled or "residual" modes. Balas (ref. 2) has used the term "control spillover" for the unwanted forcing of the residual modes by the control inputs, and the term "observation spillover" for the unwanted contribution of the residual modes to the observations. If the residual modes have some natural damping, the closed-loop system would be stable in the absence of either or both spillovers. However, spillover terms are present in practice and must be considered in overall system performance. In this paper several approaches to the design of reduced order controllers for large space structures are presented and discussed. The approaches are based on LQG theory and include truncation, singular perturbation regulator and estimator, direct sensor feedback, higher order estimator, and selective suppression of some of the uncontrolled or "residual" modes. Numerical results are given for a long free-free beam. Sufficient conditions for stability in the presence of control and observation spillovers are also presented.

MATHEMATICAL MODEL OF A FREE-FREE BEAM

The normal coordinate planar equations of motion of a uniform free-free beam can be derived as (ref. 3)

$$\ddot{\theta} = \frac{1}{I} \sum_{j=1}^{m} T_j \tag{1}$$

$$\ddot{\xi}_i + \omega_{si}^2 \xi_i = \sum_{j=1}^{m} \phi'(x_{Tj}) T_j \tag{2}$$

where θ is the rigid body angle, T_j, $j = 1, 2, \ldots, m$ are the moments applied at points $x_{T1}, x_{T2}, \ldots, x_{Tm}$, ξ_i is the modal displacement for the ith mode, ω_{si} is the ith natural frequency, ϕ_i' is the slope of the ith normalized mode shape. Expressions for ϕ_i and ω_{si} are given in ref 3. The exact model consists of i = 1, 2, 3,..., ; however, it is customary in practice to truncate the model at $i = n_e$. The resulting model will be called the "evaluation model."

Denoting

$$x_e = (\theta, \dot{\theta}, \xi_1, \xi_2, \ldots \xi_{n_e}, \dot{\xi}_1, \dot{\xi}_2, \ldots, \dot{\xi}_{n_e}) \tag{3}$$

The state equations can be written as

$$\dot{x}_e = A_e x_e + B_e T \tag{4}$$

where the appropriately dimensioned matrices A_e and B_e, can be obtained from equations (1) and (2). The order of the system is $n = 2(\eta_e + 1)$. Although only torque generators (e.g., control moment gyros or annular momentum control devices (ref. 4)) are considered in this paper for actuation, force generators can be easily included.

Since all physical systems have some damping, damping terms should be included in the model to make it more realistic. This can be (approximately) done by adding the term $2\rho_{si}\omega_{si}\dot{\xi}_i$ to the left hand side of equation (2).

The order of the evaluation model would generally be too high to permit its direct use in the design process. Therefore, a "design" model consisting of a smaller number of structural modes must be used. The state equations for the "design" model would have exactly the same form as equation (4) with n_e replaced by n_d, the number of structural modes retained.

Table I gives the natural frequencies and assumed damping ratios for an example beam which will be used for evaluating the control laws. The "evaluation" model consists of one rigid body (rotation) mode and the first seven bending modes.

Sensor Equations

The sensors considered in this paper are assumed to consist of attitude sensors and attitude rate sensors (e.g., star trackers and rate gyros). The star trackers are placed at locations ℓ_{si} and the rate gyros are placed at locations ℓ_{gi} on the beam ($i = 1, 2, \ldots n_s$). The observation equation is:

$$y = C_e x_e + w \tag{5}$$

where

$$C_e = \left[C_{e1}{}^T, C_{e2}{}^T, \ldots C_{en_s}{}^T \right]^T \tag{6}$$

$$C_{ei} = \begin{bmatrix} 1 & 0 & \phi_1'(\ell_{si}) \ldots\ldots \phi_{n_e}'(\ell_{si}) & 0 & 0 \ldots\ldots & 0 \\ 0 & 1 & 0 & \ldots\ldots & 0 & \phi_1'(\ell_{gi}) \ldots\ldots \phi_{n_e}'(\ell_{gi}) \end{bmatrix} \tag{7}$$

The dimension of y is $\ell = 2n_s$, and w is the ℓ-vector sensor noise.

Controllability and Observability

An examination of the state space equations using well-known results for multivariable systems reveals the following properties:

1) The evaluation model with torque actuators is controllable if and only if at least one actuator is placed at a nonzero-slope point of ϕ_i, $i = 1, 2, \ldots, n_e$.
2) The evaluation model with attitude and rate sensors is observable if and only if at least one attitude sensor is placed at a nonzero-slope point of ϕ_i. The system is not observable with rate sensors alone. (All modes except the rigid body mode are observable with rate sensors alone as long as at least one rate sensor is placed at a nonzero-slope point of the mode shapes.)

CONTROL SYSTEMS DESIGN

Steady-state Linear-Quadratic-Gaussian (LQG) control theory is used as the basic design tool in this paper. As the first step a complete LQG design was obtained for the full 7-mode evaluation model with one actuator and two sensors (one attitude and one rate). Nominal values of state and control weighting matrices Q and R, and of input and observation noise covariance matrices V and W were arbitrarily chosen in such a manner that the closed-loop rigid body frequency was greater than 0.1 rad/sec, and the structural mode damping ratios were all greater than 0.05. With this system as a baseline, approaches to the design of reduced order controllers were investigated using a "design" model which was assumed to consist of one rigid body mode and the first two structural modes. The evaluation and design models were discussed previously.

Design Based on Truncation

This is the most straightforward control system design (figure 2) and uses only the "design" model for obtaining the regulator and estimator gains. The equations are:

$$\dot{x}_d = A_d x_d + B_d u + v_d \tag{8}$$

$$\dot{x}_r = A_r x_r + B_r u + v_r \tag{9}$$

$$u = G\hat{x}_d \tag{10}$$

$$\dot{\hat{x}}_d = A_d\hat{x}_d + B_d u + K(y - C_d\hat{x}_d) \tag{11}$$

$$y = C_e x_e + w = C_d x_d + C_r x_r + w \tag{12}$$

where u = T, and subscript "d" denotes the "design" model.

In these equations, $B_r u$ is the "control spillover" term, and $C_r x_r$ is the "observation spillover" term. G is the constant feedback gain matrix and K is the constant Kalman gain matrix obtained by minimizing

$$J = \lim_{T_f \to \infty} \frac{1}{T} \mathcal{E} \int_0^{T_f} (x_d{}^T Q_d x_d + u^T R u)\, dt \tag{13}$$

for the "design" model. In the absence of either or both spillover terms the system would be stable. The system would also be stable for sufficiently low values of spillover terms, obtained by using small G and K. Starting with nominal values of Q, R, V, and W, R was increased successively by a factor of 10 at each step, and the closed-loop eigenvalues were obtained for the case with $n_d = 2$, $n_s = 1$. Figure 3 shows a plot of the rigid body closed-loop frequency (ω_{rb}) corresponding to the regulator, and damping ratios of structural modes. Residual modes 4 and 5, which are unstable in the beginning, become stable after step 3. For obtaining a minimum damping ratio of 0.0045 for the residual modes, R of step 4 must be used, which corresponds to ω_{rb} of 0.06 rad/sec and damping ratios $\rho_{s1} = 0.13$ and $\rho_{s2} = 0.035$. Figure 4 shows the reduction of control spillover $\|B_r G\|_s$ as $\|R\|_s$ increases ($\|H\|_s \triangleq$ (maximum eigenvalue of $H^T H)^{\frac{1}{2}}$, denotes the spectral norm of a real matrix H). Similar results are obtained by increasing W. An appropriate design procedure, therefore, could consist of alternately increasing R and W until desired closed-loop characteristics are obtained.

The weights in the nominal design were selected to be suitable for the case where the requirement on closed-loop rigid body bandwidth (ω_{rb}) is not high, and gave a ω_{rb} of about 0.1 rad/sec. In practice the required rigid body bandwidth would be determined by the pointing accuracy requirement, noise parameters, and the torque limit for the actuator. ω_{rb} can be increased by increasing the weight Q_{11} on rigid body attitude. However, stability of the system must be ensured while using the higher feedback gains that result.

Use of Low Pass Filters

A standard method of designing spacecraft attitude control systems has been to include a bank of low pass filters (LPF) in the observation circuit for reducing the observation spillover. Therefore, low pass filters of the type $\omega_{LP}^r/(s + \omega_{LP})^r$ were included in the design based on truncation and the order r of the filters as well as their cut off frequency ω_{LP} was varied from 0.1 to 120 rad/sec. However, except for large values of ω_{LP} (>100 rad/sec), the system which was stable without LPF showed instability.

It was found to be impossible to stabilize an unstable design using LPF. This can be attributed to the excessive phase lag due to the presence of both LPF and state estimator.

Singular Perturbation Regulator and Estimator

In this method, the residual system is assumed to be the "fast" subsystem. Considering only the regulator problem, it has been shown (ref. 5) that regulator design for the "slow" subsystem gives a first order approximation to the true optimal performance. The LSS model, however, does not really satisfy the assumptions used in singular perturbation regulator (SPR) design. The system has very low damping, and the residual subsystem cannot truly be described as the "fast" subsystem. However, it was hoped that this approach would redistribute the weights in the R matrix (for more than one actuator) and would result in some improvement in the performance. This approach has been used in ref. 6 and has been reported to give satisfactory performance despite the fact that two time-scales do not differ significantly. Considering only the regulator problem with no cross penalty on x_r and x_d, and ignoring noise terms,

$$J = \int_0^\infty (x_d^T Q_d x_d + x_r^T Q_r x_r + u^T R u)\, dt \qquad (14)$$

The slow subsystem is obtained by making $\dot{x}_r = 0$.

$$x_r = - A_r^{-1} B_r u \qquad (15)$$

Substituting for x_r in eqn. (14), the effective weighting matrix for u becomes:

$$R_{eff} = R + B_r^T A_r^{-T} Q_r A_r^{-1} B_r \qquad (16)$$

The feedback control gain G acting on x_d is then designed in the usual manner. The resulting regulator is a first-order approximation to the optimal regulator (ref. 5).

The suboptimal controller can be obtained by combining the above regulator with a singular perturbation estimator (SPE). This can be obtained by noting that

$$y \simeq C_d x_d - C_r A_r^{-1} B_r u + w - C_r A_r^{-1} v_r \qquad (17)$$

where v_r is the input noise associated with the x_r-equation. The resulting observation noise covariance is:

$$W_e = W + C_r A_r^{-1} V_r A_r^{-T} C_r^T \qquad (18)$$

Also, v_r may be correlated with v_d, in which case the "slow" Kalman filter should be designed taking into account the fact that input and observation noise are correlated. The resulting estimate $\hat{x}_d$ is a first-order approximation (in the sense of error covariance) to the optimal estimate (ref. 7).

This type of suboptimal controller was designed using the design model. The quantities Q_r and V_r were varied and the closed-loop eigenvalues were computed. Since the design used only one actuator and two sensors, the difference between truncation design and singular perturbation design was found to be insignificant. It was, however, expected that a significant difference would result if a larger number of sensors and actuators were used.

Direct Sensor Feedback (DSF)

If the number of independent observations available is at least $2(n_d + 1)$, an estimate of x_d can be obtained by a simple algebraic operation on the observations. This estimate would, in general, be inaccurate because of observation spillover and measurement noise. This estimate can be used for control input synthesis if the sensor noise level is low, or if the structure being controlled does not require high pointing accuracy (e.g., a stabilized platform which supports other precision structures that require high pointing accuracy). In this section, it has been assumed that $(n_d + 1)$ attitude sensors and as many rate sensors are mounted at different points on the beam, so that an estimated value of x_d is:

$$\bar{x}_d = C_d^{-1} y = x_d + C_d^{-1}(C_r x_r + w) \qquad (19)$$

where it has been assumed that C_d is nonsingular.

A DSF controller was designed for $n_d = 2$, and closed loop eigenvalues were obtained for increasing values of Q_{11} which in turn increased the rigid body frequency ω_{rb}. Stability was obtained for Q_{11} less than a certain value. Figure 5 shows a comparison of a DSF controller and a controller using an estimator. The same regulator design was used for both cases. The estimator used was the one designed earlier for truncation. For the same closed-loop rigid-body frequency, ω_{rb}, the damping ratios for the DSF controller are superior to those for the controller using an estimator.

In order to fully explore the potential of singular perturbation design, successively large number of actuators were used in the DSF design using singular perturbation. The effective control weighting matrix is given by R_{eff} in eqn. (16). For making a proper comparison with the truncation method, R for truncation was made equal to diag. $(\| R_{eff} \|_s)$. R_{eff} was changed by increasing Q_r at each step. Figure 6 shows the value of ω_{rb} achieved in both cases with the constraint that the resulting DSF controller design is stable. The singular perturbation regulator is clearly superior to the truncation regulator for this case.

Use of LPF and DSF

It was stated earlier that LPF used with a state estimator failed to make the closed-loop system stable. This was attributed to the excessive phase lag caused by using both an estimator and LPF. DSF design does not have this problem since no dynamic state estimator is used. When a bank of LPF was used in the observation circuit of the DSF design, some improvement was observed in the closed-loop eigenvalues. For example, for one controlled mode ($n_d = 1$), the maximum attainable ω_{rb} (obtained by increasing Q_{11}) for a stable design was 0.87 rad/sec, and it increased to 1.002 rad/sec with first order LPF with a cutoff frequency of 2 rad/sec. For $n_d = 2$, the maximum attainable ω_{rb} (with stability) increased from 2.56 rad/sec to 3.98 rad/sec using first order LPF with a cutoff at 4.4 rad/sec. Use of higher order LPF does not necessarily improve the performance over first order LPF.

Use of Higher Order Estimator

Consider the estimator which estimates rigid body attitude and rate and the first n_c modes:

$$\dot{\hat{x}}_c = A_c\hat{x}_c + B_c u + K\,(y - C_c\hat{x}_c) \tag{20}$$

$$y = C_c x_c + C_r' x_r' + w \tag{21}$$

where x_r' is the residual state vector for "unestimated" modes. Let the controller be given by

$$u = G_1\hat{x}_c = [G \,\vdots\, 0]\begin{bmatrix}\hat{x}_d \\ \hat{x}_{c2}\end{bmatrix} \tag{22}$$

where x_{c2} is the state vector for modes which are estimated, but not used in regulator design.

Theorem 1.- The system of equations (8), (9), (20), (21), and (22) is stable if a full order estimator is used in conjunction with a reduced order regulator.

Proof.- In this case, $n_c = n_e$, and the closed-loop equations can be written as (ignoring noise terms):

$$\begin{bmatrix}\dot{x}_e \\ \dot{\varepsilon}\end{bmatrix} = \begin{bmatrix}A_e + B_eG_1 & -B_eG_1 \\ 0 & A_e - KC_e\end{bmatrix}\begin{bmatrix}x_e \\ \varepsilon\end{bmatrix} \tag{23}$$

where $\varepsilon = x_e - \hat{x}_e$ (24)

Therefore, closed loop eigenvalues are:

$$\{\lambda_{c\ell}\} = \{\lambda(A_e + B_eG_1)\} \cup \{\lambda(A_e - KC_e)\} \tag{25}$$

But

$$A_e = \begin{bmatrix}A_d & 0 \\ 0 & A_r\end{bmatrix} \quad B_e = \begin{bmatrix}B_d \\ B_r\end{bmatrix} \quad A_e + B_eG_1 = \begin{bmatrix}A_d + B_dG & 0 \\ B_rG & A_r\end{bmatrix}$$

and $\{\lambda(A_e + B_eG_1)\} = \{\lambda(A_d + B_dG)\} \cup \{\lambda(A_r)\}$ (26)

This completes the proof.

Although it is impossible to use a full order estimator in practice, theorem 1 suggests that a higher order (than the regulator) estimator may sufficiently reduce observation spillover and make the system stable. Going back to the single actuator, two sensor cases, a truncation design was obtained with estimators estimating 2, 3, 4, 5, and 6 modes, while the 2-mode regulator remained fixed. Figure 7 shows the variation of $\| KC_r' \|_s$ as a function of n_c, the number of modes estimated. Stability was obtained for $n_c \geq 5$.

Selective Modal Suppression

While using a reduced order design model, the first few residual modes are likely to become unstable because of control and observation "spillover." It has been suggested (ref. 2) that observation spillover due to some of the residual modes can be reduced or eliminated by using such devices as phase locked loops (PLL) or comb filters. To investigate this idea, modes 3, 4, and 5 were suppressed in the "evaluation" model, and a 2-mode truncation regulator and estimator was designed for the single actuator, two sensor cases. The resulting system was stable, although some of the structural damping ratios were lower than their open-loop values. The applicability of the idea, however, remains dependent on the availability of the physical devices involved, and the extent to which they can suppress desired modes.

Sufficient Condtions for Asymptotic Stability

Control and observation "spillovers" play an important role in the stability of the system. It would be of interest of obtain the allowable limits on the spillover terms which would guarantee stability.

The controlled and residual modes are represented by equation (8) and (9) and observations are given by equation (12).

In the case of DSF, the control input is synthesized as (ignoring noise):

$$u = GC_d^{-1} y = Gx_d + GC_d^{-1} C_r x_r \tag{27}$$

In the case of a design using state estimators or LPF with DSF, the input is synthesized as

$$u = G_o z \tag{28}$$

where the n_z-dimensional compensator state vector obeys the equation

$$\dot{z} = A_f z + B_f u + Ky \tag{29}$$

In the case of LPF, $B_f = 0$, the filtered output is given by:

$$y_f = C_f z \tag{30}$$

and

$$G_o = G_f \triangleq GC_d^{-1} C_f \tag{31}$$

In the case of an observer or estimator, z is the estimate of the state vector of modes which are estimated, and $G_o = G_1$. The estimator estimates at least the modes that are controlled and possibly more.

Let (P_1, Q_1), (P_r, Q_r) be ordered pairs of $n_1 \times n_1$ and $n_r \times n_r$ positive definite symmetric matrices ($n_1 = 2n_d + 2 + n_z$, $n_r = 2(n_e - n_d)$) satisfying Lyapunov matrix equations:

$$A_1^T P_1 + P_1 A_1 = -Q_1 \tag{32}$$

$$A_r^T P_r + P_r A_r = -Q_r \tag{33}$$

where

$$A_1 = \begin{bmatrix} A_d & B_d G_o \\ KC_d & A_f + B_f G_o \end{bmatrix}$$

Let $\lambda_M(\ )$ and $\lambda_m(\ )$ denote the maximum and minimum eigenvalues of a real symmetric matrix. The following theorem gives a sufficient condition for asymptotic stability of systems using an estimator or DSF with LPF.

Theorem 2.- The closed-loop system defined by equations (8), (9), (12), (28) and (29) (ignoring noise terms) is asymptotically stable if

$$\| KC_r \|_s \ \| B_r G_o \|_s < \lambda_m(Q_1)\lambda_m(Q_r) / \left\{ 4\lambda_M(P_1)\lambda_M(P_r) \right\} \tag{34}$$

Proof.- Proof for the case with estimators has been given in ref. 8 via two methods (using vector Lyapunov functions and weighted sum of scalar Lyapunov functions). The LPF-DSF case has been shown above to fit in the same framework. The result is therefore applicable to both cases.

Theorem 3.- DSF without LPF

The closed-loop system using DSF without LPF is stable if

$$\| B_r G \|_s \ \| C_d^{-1} C_r \|_s < 1 \Bigg/ \left[2 \frac{\lambda_M(P_r)}{\lambda_m(Q_r)} \left\{ 1 + \frac{2\lambda_M(P_1) \, \| B_d G \|_s}{\lambda_m(Q_1)} \right\} \right] \tag{35}$$

Proof.- The proof can be obtained via the results of Grujic and Siljak (ref. 9) which use the vector Lyapunov function approach. The steps are very similar to those used for proving Theorem 2 (ref. 8) with minor differences, because of the term $B_r G C_d^{-1} C_r x_r$ appearing in the residual equation. The proof also uses the fact that, given two matrices A and B,

$$\| AB\ x \| \leq \| A \|_s \| B\ x \| \leq \| A \|_s \ \| B \|_s \ \| x \|$$

For a given design, these conditions would be useful in determining the region in the parameter space of spillover matrices in which the system is stable, or for determining the stable regions in the space of regulator and estimator gains. For example, it would be useful for predicting the accuracy requirement for PLL or notch filters used in selective model suppression.

CONCLUSIONS

Using LQG theory as the basic design tool, several methods have been presented and discussed for the control of large space structures. The model used for a representative LSS was that of a uniform free-free beam since it is well-defined, simple, and retains the key dynamic characteristics of LSS. Most of the practical LSS would probably be modeled using the finite element method. Controller design methods described in this paper can also be used for finite element models.

It has been shown that the system considered can be satisfactorily controlled using a single torque generator and one attitude and one rate sensor. However, the Direct Sensor Feedback (DSF) method appears to be superior when a sufficient number of sensors are available and (1) the sensor noise level is low, or (2) the pointing requirement on the LSS to be controlled is not very stringent. The situation in (2) may arise, for example, when the LSS is a stabilized platform used for supporting a number of other precision pointed structures. Use of singular perturbation in the regulator design for multiple actuators was found to be superior to the truncation method because of its ability to properly distribute the control weighting elements. Use of low pass filters (LPF) with DSF was found to be advantageous.

Other methods which were found to be potentially useful were use of higher order estimators and selective model suppression. Given the "evaluation" model and pointing accuracy requirements, controller design for an actual LSS would probably begin with the determination of minimum required ω_{rb} based on the sensor and actuator noise parameters. The next step would be to determine the structural modes that have to be actively controlled. An appropriate design method could then be selected depending on the mission, the modes to be actively controlled, number of sensors and actuators, etc. This design would be based on the apriori knowledge of the system parameters, which would in general be inaccurate. Therefore, sensitivity to modeling inaccuracies and parameter variation is an important part of the controller design problem. Sensitivity studies, parameter estimation and possibly adaptive control are important parts of continuing research in this direction.

REFERENCES

1. Outlook for Space: NASA SP-386 and SP-387, January 1976.

2. Balas, Mark J.: Active Control of Flexible Systems. Symposium on Dynamics and Control of Large Flexible Spacecraft. Blacksburg, VA, June 1977.

3. Greensite, A. L.: Analysis and Design of Space Vehicle Flight Control Systems, Spartan Books, 1970.

4. Anderson, W. W.; and Groom, N. J.: The Annular Momentum Control Device (AMCD) and Potential Applications, NASA TN D-7866, March 1975.

5. Chow, J. H.; and Kokotovic, P. V.: A Decomposition of Near Optimum Regulators for Systems with Slow and Fast Modes. IEEE Trans. Auto. Control, Vol AC-21, No. 5, October 1976.

6. Sesak, J. R.: Sensitivity Constrained Linear Optimal Control: Analysis and Synthesis. Ph.D. Thesis, Dept. of Electrical Eng., University of Wisconsin, Madison 1974.

7. Haddad, A. H.: Linear Filtering of Singularly Perturbed Systems. IEEE Trans. Auto. Control, Vol. AC-21, No. 4, August 1976.

8. Joshi, S. M.; and Groom, N. J.: Stability Bounds for the Control of Large Space Structures. Submitted to AIAA Journal of Guidance and Control.

9. Grujic, Lj. T.; and Siljak, D. D.: Asymptotic Stability and Instability of Large Scale Systems. IEEE Trans. Auto. Control, Vol. AC-18, No. 6, December 1973

TABLE I.

Natural Frequencies and Damping Ratios for Example Beam

Mode No.	1	2	3	4	5	6	7
ρ_{si}	.005	.0055	.00605	.00665	.00732	.00805	.00886
ω_{si} (rad/sec)	.428	1.172	2.297	3.797	5.672	7.921	10.546

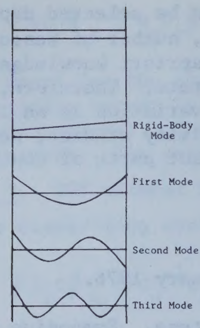

Rigid-Body Mode
First Mode
Second Mode
Third Mode

Figure 1.- Mode shapes of uniform free-free beam.

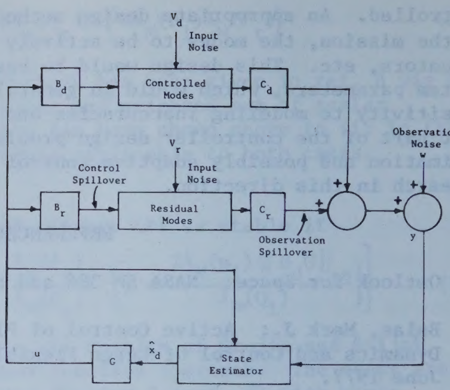

v_d
Input Noise
B_d
Controlled Modes
C_d
Observation Noise
w
v_r
Control Spillover
Input Noise
B_r
Residual Modes
C_r
+
+
+
+
Observation Spillover
y
u
G
$\hat{x}_d$
State Estimator

Figure 2.- Controller design based on truncation.

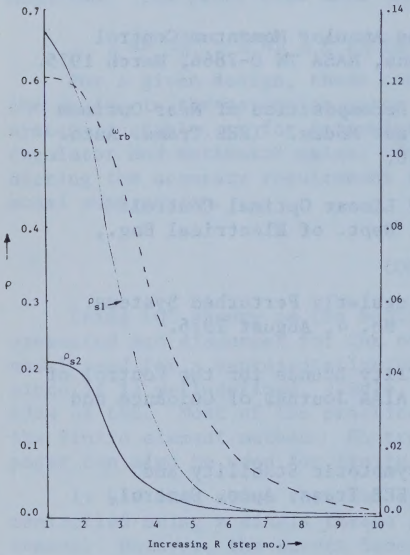


Figure 3a.- Variation of ω_{rb} and ρ_{si} with R

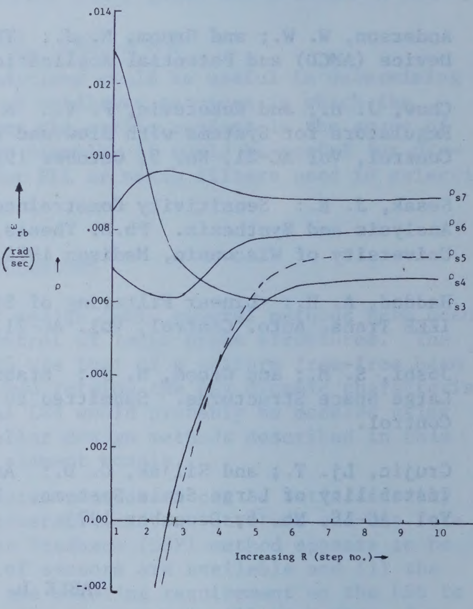


Figure 3b.- Variation of ρ_{si} with R

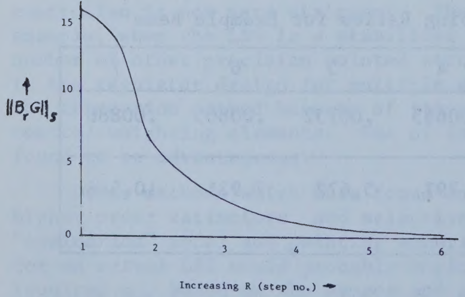


Figure 4.- Variation of control spillover with R.

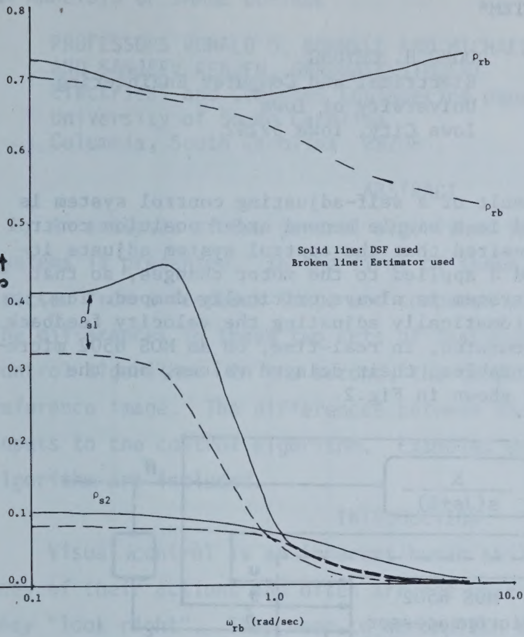


Figure 5a.- Comparison of DSF and controller using estimator.

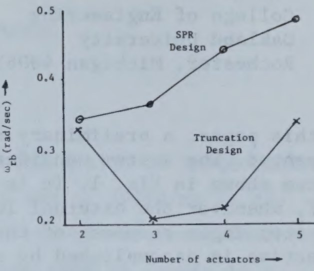


Figure 6.- Comparison of SPR and truncation designs.

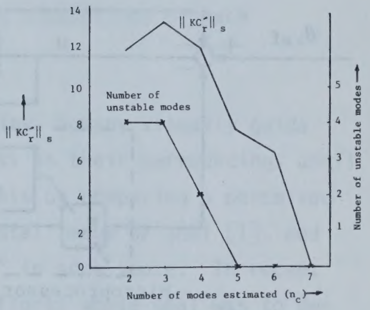


Figure 7.- Variation of observation spillover with n_c.

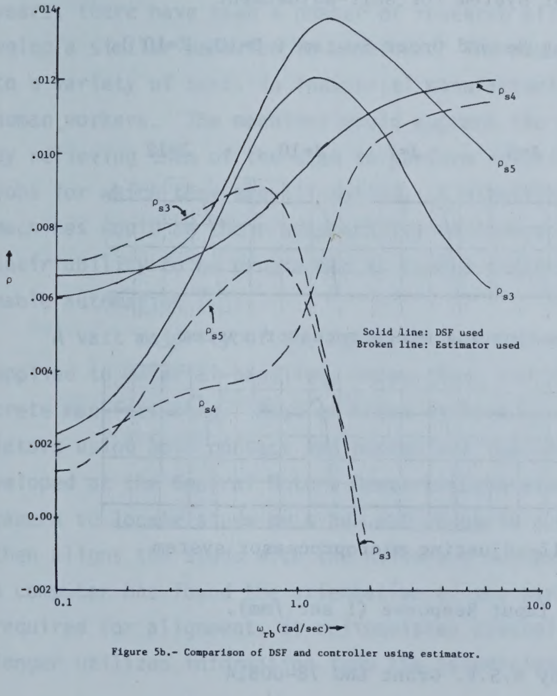


Figure 5b.- Comparison of DSF and controller using estimator.

ON A SELF-ADJUSTING CONTROL SYSTEM*

NAN K. LOH
College of Engineering
Oakland University
Rochester, Michigan 48063

DONG H. CHYUNG
Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242

In this paper, a preliminary result of a self-adjusting control system is presented. The system considered is a simple second order position control system shown in Fig. 1. It is desired that the control system adjusts itself, whenever the external load J applied to the motor changes, so that the step input response of the system is always critically damped. The objective is accomplished by automatically adjusting the velocity feedback constant K_v. The scheme is implemented, in real-time, on an MOS 6502 microprocessor by using the state variables, their delayed values, and the control variable. The result is shown in Fig.2.

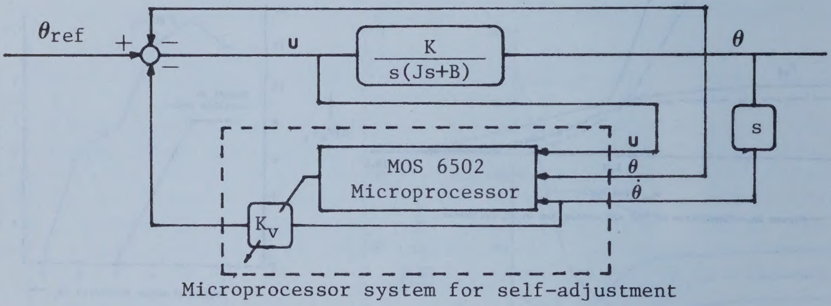
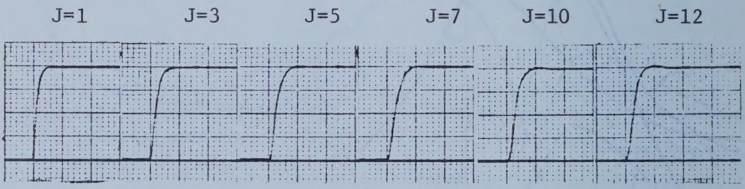
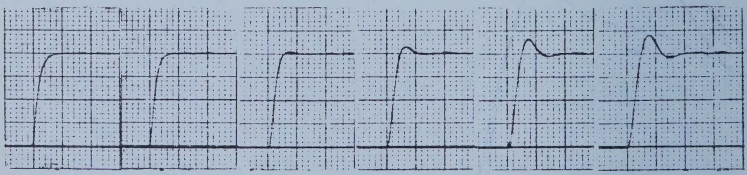

θ_{ref}
+
−
−
u
$\frac{K}{s(Js+B)}$
θ
s
u
θ
$\dot{\theta}$
MOS 6502 Microprocessor
K_v
Microprocessor system for self-adjustment

Fig.1 Self-Adjusting Second Order System (B=10, K=10).



(a) With self-adjusting microprocessor system



(b) Without self-adjusting microprocessor system

Fig.2 Step Input Response (1 sec./mm).

---

*This research is supported by N.S.F. Grant ENG 78-00814

AN ANALYSIS OF IMAGE CONTROL

PROFESSORS RONALD D. BONNELL AND MICHAEL N. HUHNS
AND SANJEEV RENJEN, GRADUATE STUDENT
Electrical and Computer Engineering Department
University of South Carolina
Columbia, South Carolina 29208

ABSTRACT

An analysis of two general algorithms for the design of image control systems is presented. In the first of these, a set of features is extracted from a feedback image and compared to a set of reference features. The differences in these two sets of features are used as inputs to a control algorithm. In the second, the feedback image is compared to a reference image. The differences between these two images are used as inputs to the control algorithm. Examples and simulations of each algorithm are included.

INTRODUCTION

Visual control is an inherent human ability: humans visually guide many of their actions and often arrange objects in their surroundings until they "look right". They seem to accomplish this by comparing a perceived image of their surroundings with a desired mental image or goal [1], and then make adjustments until the images "match" in some sense. In recent years, there have been a number of research efforts whose goal was to develop a similar behavior in machines. The machines could then be applied to a variety of tasks in industrial manufacturing that currently require human workers. The machines would augment the capabilities of the workers by relieving them of the need to perform relatively low-level, repetitive jobs for which they are ill-suited. A significant feature of these machines would be their adaptability to changes in the environment and their ability to be programmed to handle different tasks, termed programmable automation [2].

A vast majority of the research in programmable automation has been applied to material-handling, inspection, and assembly operations in discrete manufacturing. Most of these systems consist of hand-eye manipulators using both contact and noncontact feedback sensors. A system developed at the General Motors Research Laboratories uses a television camera to locate studs on a hub,and holes in a wheel [3]. A manipulator then aligns the studs with the holes and mounts the wheel on the hub. Once a computer has found the orientation of the parts and calculated the motion required for alignment, it relinquishes control to the manipulator and no longer utilizes information from the television camera. The wheel-mounting system thus works in an open-loop mode.

An Hitachi hand-eye system [4] uses a vidicon camera and a suction-cup mechanical hand interfaced to a minicomputer. Camera images are processed to find the location and orientation of a block on a conveyor. The block is then moved by the hand under the open-loop control of the computer. A similar system has been developed by Heginbotham et al. [5].

At the Stanford Research Institute, Agin and Duda [6,7] and Rosen et al. [8] have developed a system for detecting the position and orientation of a randomly-placed part. Two cameras are used to perform the visual recognition and a Unimate industrial robot is used to acquire and move the part. The primary goal of this research effort has been the accurate identification of unknown parts, rather than the movement of the parts. The results are applicable to discrete manufacturing and involve only open-loop control.

Earlier work by Pringle [9] consisted of a similar system to stack blocks. This system employed a scheme of successive approximations to control the motions of the blocks. The control technique, however, was heuristic.

Other significant applications of image processing within a control system environment have been to correlation navigation systems [10-12]. In these systems, an image of the surroundings of a vehicle is compared to an ideal map or image by means of two-dimensional correlation techniques. The results of this correlation specify positional information which is then used to navigate the vehicle.

The above automation and navigation applications have been particularly configured for discrete systems.

The above control systems involving feedback image information have been applied to discrete processes. Image control systems are applicable not only to these problems, but also to the control of continuous manufacturing processes. Systems utilizing visual feedback can be configured for solving either the tracking problem or the regulator problem of continuous control. The resultant systems have the potential to be more accurate, reliable, and adaptable than conventional systems.

ANALYSIS

Image control consists of a combination of image processing and control theory, in which a control system is designed using images for the feedback information. A general configuration for this type of control system is shown in Figure 1. The plant to be controlled is assumed to generate an image that contains sufficient information, at least partially, to

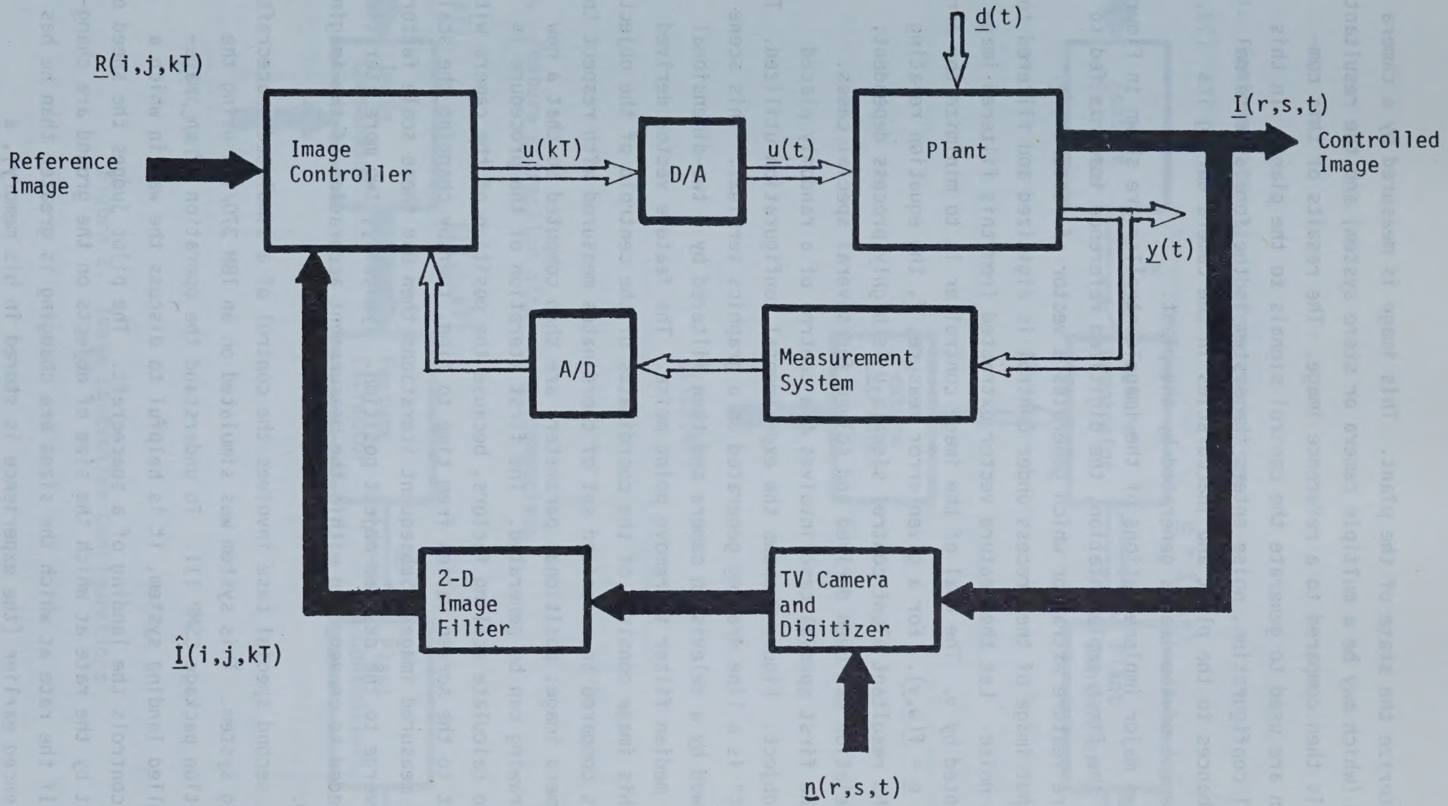

R(i,j,kT)
Reference Image
Image Controller
u(kT)
D/A
u(t)
d(t)
Plant
I(r,s,t)
Controlled Image
y(t)
Measurement System
A/D
2-D Image Filter
TV Camera and Digitizer
Î(i,j,kT)
n(r,s,t)

Figure 1. General Configuration for an Image Control System

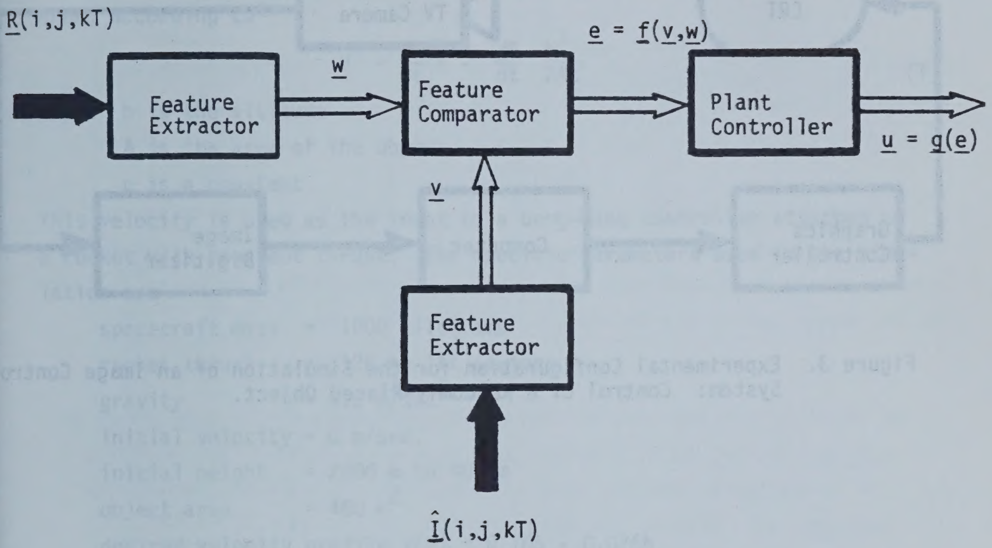
characterize the state of the plant. This image is measured by a camera system (which may be a multiple camera or stero system) and the resultant image is then compared to a reference image. The results of this comparison are used to generate the control signals to the plant. In this general configuration, noise enters the system in the form of external disturbances to the plant and inaccuracies in the camera during its measurement of the images generated by the plant.

Two major implementations of the image controller are shown in Figure 2. In the first implementation, the digitized reference image is fed to a software feature extractor which generates a vector of features, $\underline{w}$. An output image of the process under control is digitized and filtered to remove noise. Let the feature vector extracted from this filtered image be denoted by $\underline{v}$. The goal of the image controller is to minimize the error vector $\underline{e} = \underline{f}(\underline{w},\underline{v})$. For a given error measure, $\underline{f}$, the equation relating $\underline{e}$ to the resultant plant control signals $\underline{u}$ is highly process dependent; this relation has been derived and solved for several special cases.

The first special case involves the control of a randomly placed known object. Figure 3 shows the experimental configuration utilized. The "object" is a line drawing generated on a graphics terminal. This scene is viewed by a television camera and then filtered by a two-dimensional Tukey median filter to remove point noise. The feature vector derived from this image consists of the coordinates of the centroid of the object. This is compared to a desired set of coordinates measured with respect to the camera image; positional parameters are then computed so that a new line drawing can be generated. The first iteration of the procedure is used to calculate scaling factors, because the position of the camera with respect to the screen varies from time to time, thereby changing the scale of the measured image. Subsequent iterations then use these scale factors to converge to the desired object position. Typically, two more iterations are needed to converge to within the measurement accuracies of the imaging system.

A second special case involves the control of a simulated spacecraft landing system. This system was simulated on an IBM 370/168 using the simulation package CSMP III. To understand the operation of an image-controlled landing system, it is helpful to discuss the way in which a pilot controls the landing of a spacecraft. The pilot judges the speed of descent by the rate at which the size of objects on the ground are changing. If the rate at which the sizes are changing is greater than he has experienced earlier (the experience is stored in his memory), a

a. Feature extraction before comparison


$\underline{R}(i,j,kT)$
Feature Extractor
$\underline{w}$
Feature Comparator
$\underline{e} = \underline{f}(\underline{v},\underline{w})$
Plant Controller
$\underline{u} = \underline{g}(\underline{e})$
$\underline{v}$
Feature Extractor
$\hat{\underline{I}}(i,j,kT)$

b. Feature extraction after image comparison

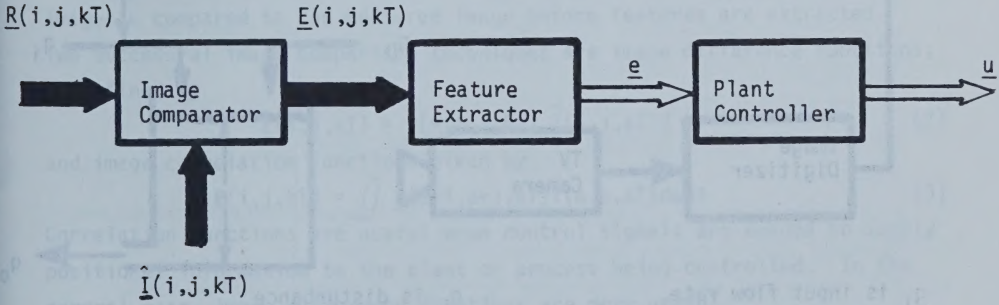

$\underline{R}(i,j,kT)$
$\underline{E}(i,j,kT)$
Image Comparator
Feature Extractor
$\underline{e}$
Plant Controller
$\underline{u}$
$\hat{\underline{I}}(i,j,kT)$

Figure 2. Image Controller Configurations

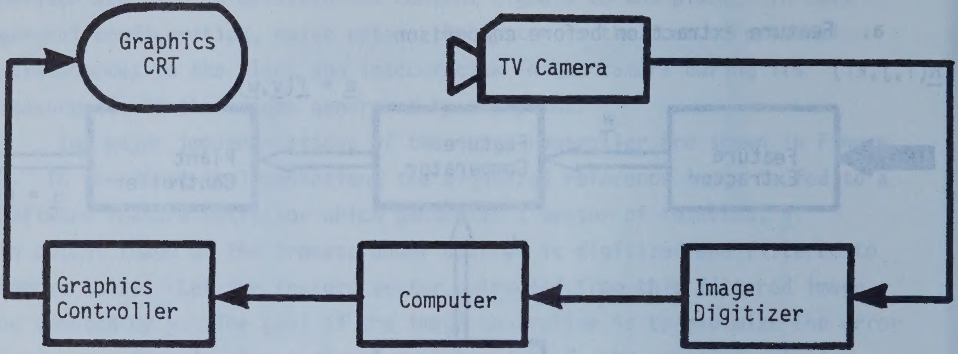

Graphics CRT
TV Camera
Graphics Controller
Computer
Image Digitizer

Figure 3. Experimental Configuration for the Simulation of an Image Control System: Control of a Randomly-Placed Object.

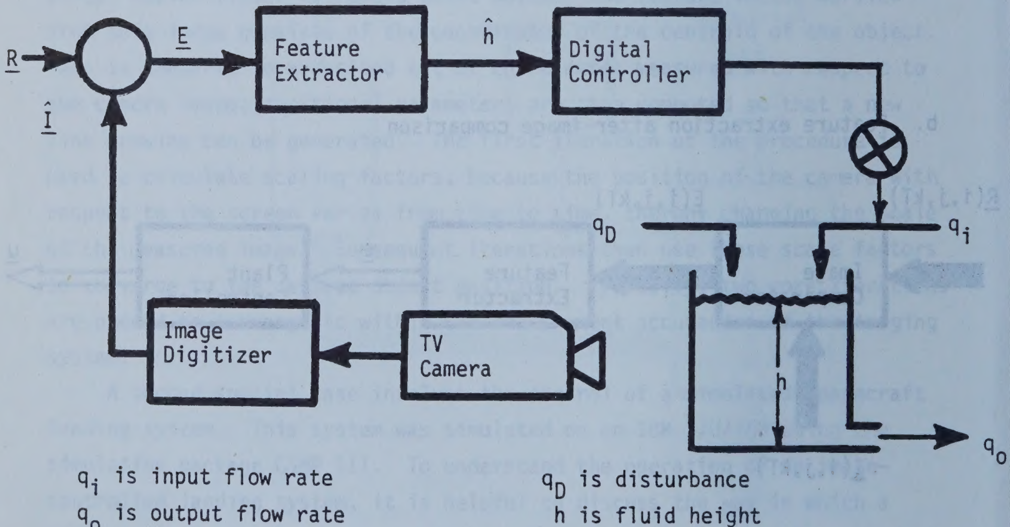

R
E
Î
Feature Extractor
ĥ
Digital Controller
q_D
q_i
h
q_o
Image Digitizer
TV Camera
q_i is input flow rate
q_o is output flow rate
q_D is disturbance
h is fluid height

Figure 4. Image Control System for the Noncontact Control of the Liquid Level in a Tank.

deceleration rocket is ignited until the landing rate appears correct.

A similar scheme is used for the image controller. A camera attached to the spacecraft views a known object on the ground. A computer digitizes this image and measures the area of the object. The velocity of the spacecraft at any instant is determined by finding the rate of change of this area, according to

$$\hat{V} = \frac{dh}{dt} = - \frac{dA}{dt} \frac{h^3}{2Ac} \qquad (1)$$

where h is the altitude
A is the area of the object
c is a constant

This velocity is used as the input to a bang-bang controller attached to a rocket with constant thrust. The specific parameters used in the simulation are

spacecraft mass = 1000 kilograms
rocket thrust = 12K to 15K newtons
gravity = 9.8 m/sec^2
initial velocity = 0 m/sec.
initial height = 2000 m to 4000m
object area = 400 m^2
desired velocity profile V(h) = 0.105 - 0.026h

The landing site image is sampled once each second.

A number of simulations were made for different thrusts, initial heights, and velocity profiles. In each case, the spacecraft tracks the landing profile closely and lands at approximately 1.5 meters/second.

In the second controller configuration (Figure 2.b) the reference image is compared to the measured image before features are extracted. Two successful image comparison techniques are image difference functions, $\underline{f}$, defined by

$$\underline{E}(i,j,kT) = \underline{f}[\underline{R}(i,j,kT),\hat{I}(i,j,kT)] \qquad (2)$$

and image correlation functions given by

$$C(i,j,kT) = \iint_{-\infty}^{\infty} \underline{R}(\alpha+i,\beta+j,kT)\hat{I}(\alpha,\beta,kT)d\alpha d\beta \qquad (3)$$

Correlation functions are useful when control signals are needed to supply positional information to the plant or process being controlled. In the general case, image difference functions are more useful.

A system, utilizing an image difference function, has been used to control the liquid level in a tank. Figure 4 shows the experimental configuration and the parameters involved. The differential equation relating the height of the liquid to the input and output flow rates is

$$A_c \frac{dh}{dt} = q_i + q_d - A_p \sqrt{2gh} \tag{4}$$

In the control algorithm implemented, the feature extracted from the difference image is the area, A_e. The goal is to minimize this area. The zero-order control equation of the tank is

$$q_i = q_{io} - kA_e \tag{5}$$

where q_{io} is the steady-state input flow rate. This configuration was simulated using CSMP III and the results indicated a stable system with negligible errors and a rapid response to disturbances and input changes.

CONCLUSION

Image control has the potential to be applicable to many facets of industrial automation, including both continuous and discrete processes. However, much additional work remains in deriving general image controllers before image control can be widely applied.

REFERENCES

1. M. Minsky, "A Framework for Representing Knowledge", The Psychology of Computer Vision, P. H. Winston, ed., McGraw-Hill, 1975, pp. 211-277.
2. C.A. Rosen and D. Nitzan, "Use of Sensors in Programmable Automation", Computer, Vol. 10, No. 12, December 1977, pp. 12-23.
3. S.L. Rossol, J. T. Olsztyn, R. Dewar, and N.R. Lewis, "An Application of Computer Vision to Automatic Wheel Mounting", Proc. International Joint Conference on Pattern Recognition, Washington, D. C., October 1973.
4. "Hitachi Hand-Eye System", Hitachi Review, Vol. 22, No. 9, pp.362-365.
5. W.B. Heginbotham, P.W. Kitchin, and A. Pugh, "Visual Feedback Applied to Programmable Assembly Machines", Proc. Second Int. Symp. on Industrial Robots, Chicago, 1972, pp. 77-88.
6. J.Agin and R.O. Duda, "SRI Vision Research for Advanced Industrial Automation", Proc. Second USA-Japan Computer Conference, 1975, pp. 113-117.
7. T.O. Binford et al., "Exploratory Study of Computer-Integrated Assembly Systems", NSF Progress Report, Sept. 1974 to Nov. 1975.
8. C.A. Rosen, et al., "Exploratory Research in Advance Automation", Reports 1 to 5, Stanford Research Institute (under NSF Grant), Dec. 1973 to January 1976.
9. K.K. Pringle, "Visual Perception by a Computer", Ch. 13 in Automatic Interpretation and Classification of Images, (A. Grasselli, ed.) Academic Press, New York, 1969, pp. 277-284.

10. S.A. Dudani, "Scene Model for Image Based Guidance Systems", Seventh Annual Automatic Pattern Recognition Symposium Proc., EAI, Washington, D.C., May 1977, pp. 230-234.
11. J.J. Hwang and J.C. Hung, "A Concept of Landmark Navigation by Perspective Projection", Proc. Southeastcon 1978, April 1978, pp. 113-120.
12. R.L. Gutman, "Course Quantization Techniques for Correlation Navigation", Proc. 1977 IEEE Conf. Decision and Control, New Orleans, La., Dec. 1977, pp. 1199-1202.

THE SADDLE SET CONCEPT AND ITS APPLICATION IN ADAPTIVE OPTIMAL CONTROL

DR. ELI FOGEL
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

The difficulties in formulating the optimal control problem in the game theoretic setting is discussed. Counter examples demonstrate that commonly used procedures may yield suboptimal control. The effect of incorporating an estimator in the control loop is discussed. Subsequently, a control procedure is presented which performs better than previous controllers. Calculating the control involves a solution of a minimax problem. To facilitate the solution, the concept of a "saddle set" is introduced and thus a simple algorithm is deviced to solve the minimax problem at each control step.

I. INTRODUCTION

The most common approach to the problem of optimal control in face of uncertainties is the stochastic approach, e.g., [1-5]. Although this may be a natural way to handle noise in the system, this approach is artificial when applied to constant uncertainties in the system. In particular, the assignment of statistics to the constant unknown parameters of the system to be controlled [4, 17-18] is difficult to justify.

On the other hand, membership set constraints on system parameters (and on noise) are usually available or can be estimated [29]. Such specifications may be handled via game theory [6-8] or sensitivity analysis [9-10]. In this paper we combine the two approaches to derive an "optimal" control algorithm.

The game theoretic approach poses a difficult conceptual problem - what is a "good" game criteria which reflects the fact that "nature" (the uncertainties generator) is not a consious player. We shall elaborate on this point in section II. Although the difficulties discussed in section II are not resolved here, a control procedure is presented in section III which partially alleviates these difficulties. This control procedure involves the solution of a complex min-max problem.

To enable the solution of the min-max problem, the concept of the "saddle set" introduced. This enables the construction of a simple fast minimaximization algorithm to be discussed in section IV.

The excellent performance of the presented algorithm is demonstrated by simulation.

II. THE CHOICE OF THE CRITERION

Let the state equation be

$$X_{k+1} = f\ (X_k,\ U_k,\ \omega_k,\ k) \quad k = 0,1,\ldots \tag{1}$$

and the measurement equation

$$Y_k = h\ (X_k,\ U_k,\ k) \tag{2}$$

where $X_k \in \mathcal{R}^n$, $U_k \in \mathcal{R}^m$ and $Y_k \in \mathcal{R}^e$ are the state control and output respectively. $w_k \in \mathcal{W}_k$ and $v_k \in \mathcal{V}_k$ are the uncertainties associated with each equation, script letters denoting membership set. A performance cost is associated with (1), given by the scalar

$$J = \sum_{i=1}^{N+1} J_i\,(X_i, U_{i-1}) \tag{3}$$

We shall use the notation U_i^j to denote the sequence $\{U_k\}_{k=i}^{j}$. The purpose of U_0^N is to minimize the cost function. Since this objective is unachievable due to W_0^{N-1}, a performance criterion is defined:

Definition $\bar{J}: \mathcal{R}^1 \to \mathcal{R}^1$ is the performance criterion of the control problem (1),(3) if it satisfies :
(a) $\bar{J}(J)$ is not function of a particular realization of W_0^{N-1} and
(b) $\bar{J}(J) = J$ *if* $\mathcal{W}_k = w_k$ $(\{w_k\}$ are known.)

In the game theory literature the choice usually is

$$\bar{J}(J) = \max_{W_0^N\ V_0^{N-1}} J(\cdot) \tag{4}$$

where maximization is over the membership sets. Thus, the control is calculated via the solution of

$$\min_{U_0^N}\ \max_{W_0^N\ V_0^N}\ J(X_o, U_0^N, W_0^N V_0^N)^{*} \tag{5}$$

Dynamic Programming: It is almost a folklore in the control society to apply dynamic programming to the solution of optimal control problems. This technique stems from Bellman's "principle" of optimality [16]. In our case "dynamic programming" is reflected in the solution of

$$\min_{u_0}\ \max_{w_0 v_0}\ \min_{u_1}\ \max_{w_1 v_1} \ldots \min_{u_N}\ \max_{w_N v_N}\ J(X_o, U^N, W^N, V^N) \tag{6}$$

Let the solution of (5) be $J(\bar{U}^N, \bar{W}^N, \bar{V}^N)$ and the solution of (6) be $J(\tilde{U}^N, \tilde{W}^N, \tilde{V}^N)$. However, it is well known that

$$\min_u \max_w J(u, w) \geqslant \max_w \min_u J(u, w) \tag{7}$$

with equality if (sufficient condition) J is convex in u and concave in w. Thus, it follows that

Proposition 1:

$$J(\bar{U}^N, \tilde{W}^N, \tilde{V}^N) \leqslant J(\bar{U}^N, \bar{W}^N, \bar{V}^N) \leqslant \max_{W^N V^N} J(\tilde{U}^N, W^N, V^N) \tag{8}$$

---

* It is assumed throughout the paper that min(·) and max(·) exist.

Relation (8) implies that the optimal control should not be found via dynamic programming as in (6). This argument however may be altered if the solution to (5) is a saddle set or if the "dual effect" [14,15] is taken into account. Before discussing these subjects the following definitions are given:

Definition 1: The sequence U_0^N is called policy if $U_i = \mu_i(Y_o^{i-1}, U_o^{i-1})$

Definition 2: The sequence U_0^N is called control if u_i is independent of the measurement.

Note that the set of controls is the subset of the set of policies satisfying.

$$\{U_i \mid \mu_i (Y^{i-1},U^{i-1}) = \mu_i(\bar{Y}^{i-1},\bar{U}^{i-1}) \ \forall\ Y^{i-1} \neq \bar{Y}^{i-1},\ \forall\ U^{i-1} \neq \bar{U}^{i-1} \text{ and } \forall\ i\} \tag{9}$$

Definition 3: The sequence U^N is called pure policy if it is a member of the set of policies excluding the set of controls.

Proposition 2: The optimal U_0^N is found in the set of policies.

In general, a solution to the optimal policy is extremely difficult to find. Furthermore, it is essential to include the set of controls in the set of policies since the optimal policy may be a control. It is impossible to state in general the superiority or inferiority of the optimal pure policy to the optimal control.

Dual Effect. So far we have ignored the possible effect of the control (*) sequence U_0^i on the uncertainties. If one takes into account the possible dependence of $\mathcal{W}_j$ on u_i the inequality in (8) may not be correct. Namely, if $\mathcal{W}_j = \mathcal{W}_j(U_0^N)$ and $\mathcal{V}_j = \mathcal{V}_j(U_0^N)$ it may be desirable to apply dynamic programming. For example if a proper choice of U_0^k implies reduction of the level of the uncertainty in W^N_k to zero, then one should solve

$$\min_{U_0^k} \ \max_{w_0^N} \ \min_{V_0^N \ U_{k+1}^N} J(\cdot) \tag{10}$$

rather than solving (5). This example although an extreme one demonstrates the difficulty in formulating the control problem.

Classification of the Control. In the stochastic control literature distinction is made between closed loop and open loop feedback [17]. Roughly, a closed loop control is a policy which takes into account the dependence of future uncertainties on the control. Thus, a closed loop control is a dual control. The open loop feedback is a control which is updated at each time instant to reflect possible uncertainties reduction achieved up to this time instant. It has been shown [25] that if $J = J(X_N)$ the open loop feedback performs better than the open loop control. This result, however, is not true in general. In [26] it was shown for the continuous time case, that the open loop feedback may be inferior to the open loop control, and can be shown also in the discrete case via the following example.

Example 1 Consider the system

$$X_{k+1} = A X_k + U_{k+1} \qquad X_o = 1 \qquad A \in [1,3]$$

---

*We talk about control sequence which may be a policy or a control.

A is unknown. The cost function is

$$J = X_1^2 + X_2^2$$

The open loop solution is given by

$$\min_{U_1 U_2} \max_{A \, \varepsilon \, [1,3]} J(U_1, U_2, A) = J(-3,2,1) = J(-3,2,3) = 4$$

To solve the open loop feedback U_1 is found as above, namely $U_1 = -3$. U_2 is found after applying U_1 by solving

$$\min_{U_2} \max_{A} J(-3,U_2,A) = \min \max (X_1^2 + (AX_1 + U_2)^2)$$

which yields $U_2 = -2X_1$. However, for $A = 1$ we get

$$J(-3,-2X,1) = (-2)^2 + (-2+4)^2 = 8 > J(-3,2,1) = 4.$$

Finally, we note that since nature is not a conscious player, it may be desirable to apply the open loop feedback to reflect the actual choice of W_0^k which is in general not optimal from the game point of view.

III. The Control Procedure

The difficulties discussed in the previous section are alleviated in this section by the introduction of a new control procedure.

III.1 The Sensitivity Function. By introducing a sensitivity function as the game criterion we somewhat alleviate the difficulty associated with the fact that nature is not a conscious player. The use of sensitivity function rather than the cost itself in game theory has been used earlier in [9-10]. For a cost $J(U,w)$ the sensitivity function chosen is

$$S(U,w) = J(U,w) - J(*(w),w) \tag{11}$$

where $\min_{U} J(U,w) = J(U^*(w),w)$. To demonstrate the advantage of the sensitivity function consider the following

Example 2 Let $J = (U-w)^2 + w^2$; $w \, \varepsilon \, [0,1] = \mathcal{W}$. The optimal solution to (11) is $U_s = 1/2$ $w_s = \{0,1\}$ whereas min max of $J(\cdot)$ yields $U_o = 1$ $w_o = \{0,1\}$. The plots of $J(U^*(w),w)$, $J(U_s, w)$ and $J(U_o,w)$ are given in Figure 1. The advantage of U_s to U_o is obvious.

III.2 The Control Algorithm. The sensitivity function is combined with an open loop feedback control to yield the following control algorithm to calculate U_k^*.

1. Estimation. Evaluate the membership sets

$$\mathcal{W}^N(k-1) \triangleq \mathcal{W}^N(U^{k-1}, Y^{k-1}, k-1) \tag{12}$$

2. Control. Solve

$$\min_{u_k} \max_{W^N \, \mathcal{W}^N(k-1)} \overset{*}{S}_k(u_k, \overset{*}{U}{}_{k+1}^N (W^N),W^N) \tag{13}$$

where

$$\overset{*}{S}_k = J(u_k, \overset{*}{U}{}_{k+1}^N (W^N),W^N) - J(\overset{*}{U}{}_k^N (W^N),W^N) \tag{14}$$

$\overset{*}{U}{}_k^N(W^N)$ is the optimal control sequence for a given W^N.

3. Apply $\overset{*}{u}_k$ which solves (13) and repeat (1-3) for k+1.

Proposition 3 The above algorithm constitutes a control procedure which performs at least as well as the open loop control even if $\mathcal{W}^N(i) = \mathcal{W}^N(0)$ $\forall$ i.

A Plausible Proof. For every control u_i a set of optimal trajectories is defined by $\{\overset{*}{X}{}_{i+1}^{N+1}(X_i, u_i, W^N), \overset{*}{U}{}_{i+1}^N(X_i, u_i, W^N)\}$. Let $\overset{*}{u}_i(\overset{*}{W}{}^N)$ be the solution of (13). Since $\overset{*}{W}{}^N$ is not unique [10], application of $\overset{*}{u}_i$ is equivalent to the choice of the family of optimal trajectories with the smallest worst cost. However, since $\overset{*}{W}{}^N$ is not unique and W^N may be different from any $\overset{*}{W}{}^N$, $\overset{*}{u}_{i+1}$ must be chosen using the same technique.

III.3 The Adaptive Control Problem. We apply the above algorithm to the following example

$$Y_k = \sum_{i=1}^{m} a_i Y_{k-1} + \sum_{j=0}^{m} b_j U_{k-j} + w_k, \quad b_o = 1 \tag{15}$$

where U_k is the input Y_k the output and w_k disturbance. Using the notation $\Theta^T = (b_o^k, b_1 \ldots b_m, a_1, \ldots a_n)$ and $X_k^T = [U_k, U_{k-1} \ldots U_{k-m}, Y_{k-1}, \ldots Y_{k-m}]$ we get

$$Y_k = \Theta^T X_k + w_k \tag{16}$$

The cost function considered is

$$J = \sum_{i=1}^{N} (\alpha_i Y_i^2 + \beta_i U_1^2) \tag{17}$$

$\underline{\Theta}$ is assumed to be unknown, however, a membership set Θ_k is identified as in [29] to yield the membership set

$$\underline{\Theta} \in \{(\Theta - \hat{\Theta}_k)^T P_k(\Theta - \Theta_k) < F_k\} = \Theta_k \tag{18}$$

The algorithm of section III.2 is applied where the control is calculated (Step 2) via

$$\min_{U_k} \max_{\theta \in \Theta_{k-1}} [J_k(U_k, \overset{*}{U}{}_{k+1}^N, \theta) - J_k(\overset{*}{U}{}_k^N, \theta)] = \min \max S_k(U_k, \theta) \tag{19}$$

where

$$J_k(\overset{*}{U}{}_k^N, \ ) = \min_{U_k^N} \sum_{j=k}^{N} (\alpha_j U_j^2 + \beta_j Y_j^2)$$

The sensitivity function in (19) is different from $\overline{S}_k(U_k^N, \theta) = J_k(U_k^N, \theta) - J(\overset{*}{U}{}_k^N, \theta)$ used in [9-10]. It is derived from Proposition 3 and satisfies $S_k(U_k \theta) < \overline{S}_k(U_k^N, \theta)$ $\forall$ θ. It can be shown [13] that $S(U_k, \theta)$ is given by

$$S_k(U_k, \theta) = (\alpha_k + e_1^T \Lambda_k e_1)\left(U_k - \frac{e_1^T \Lambda_k I_{-1} X_k}{\alpha_k + e_1^T \Lambda_k e_1}\right)^2 \tag{20}$$

where

$$\Lambda_i = \beta_i \theta\theta^T + (S_m + e_N \theta)^T P_{i+1} (S_m + e_N \theta^T); \ i > 0 \tag{21.a}$$

$$P_i = \Lambda_i - \frac{\Lambda_i e_1 e_1^T \Lambda_i}{\alpha_i + e_1^T \Lambda_i e_1} \quad P_{N+1} = 0 \tag{21.b}$$

and where

$$S_m = \left[\begin{array}{cc:cc} 0. & . & .0 & & 0 & \\ I & & 0 & & & \\ \hdashline & & & 0. & . & .0 \\ & 0 & & 0 & & I \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\} m+1 \\ \left.\vphantom{\begin{array}{c}0\\0\end{array}}\right\} n \end{array} \quad ; \; e_i^T = (0\ldots010\ldots0); I_j = I - e_j e_j^T$$

$$\underbrace{}_{m+1} \quad \underbrace{}_{n}$$

Although (20) is a quadratic function of U_k, a solution to (19) is not available due to the complex dependence on θ. To facilitate a solution the "saddle set" concept is introduced in the next section.

IV. THE SADDLE SET

The available methods in solving a minimax problems appear to be; (1) Locate a saddle point, if it exists, and show that it represents a global solution. (2) Find an analytic expression for the maximization step, if it exists, and then minimize. (3) Use an iterative procedure (a search) [24].

The first two methods are not relevant to the minimaximization of $S_j(u_j, \underline{\theta})$ since (1) a saddle point does not exist; and (2) analytic expressions of $S_j(u_j, \underline{\theta})$ in terms of $\underline{\theta}$ can not be found, let alone an analytic solution for the maximization step.

A direct search, although straightforward, usually requires a prohibitive amount of computation. Furthermore, since $S_j(u_j, \underline{\theta})$ is not an explicit function of $\underline{\theta}$, search techniques are not practicle. To overcome these difficulties we present the concept of the saddle set and use it to derive an algorithm which will enable the solution of (19).

The following assumptions are made.

Assumption A1: $J(u,\theta)$ is convex in u $\forall\ \underline{\theta}\ \varepsilon\ \Theta$

Assumption A2: $J(u,\underline{\theta})\ \varepsilon\ C_u^1$ (the set of differentiable functions in u,) $v\ \underline{\theta}\ \varepsilon\ \theta$

It is noted that $S(u,\theta)$ of (19) satisfies both assumptions. Let

$$(u_i^*, \Theta_i^*) \triangleq \left\{u_i^*, \theta_i^{*2}, \ldots, \theta_i^{*j}\right\}_{i=1}^{L}$$

be the set of solutions to min max $J(u,\theta)$ i.e., for each i, (u_i^*, Θ_i^*) satisfies

(a) $J \triangleq J\,(u_i^*, \theta_i^{*J}) \geq J\,(u_i^*,\theta)\ \forall\ \theta_i^{*j}\ \varepsilon\ \Theta_i^*$ and $\forall\ \theta\ \varepsilon\ \Theta$ (22)

(b) $\max J\,(u,\theta) > J\ \forall$ u not in the set $\{u_i^*\}\ \forall\ \theta\ \varepsilon\ \Theta$ (23)

Theorem 1 [10] The minimax value of $S(u,\theta)$ is attained at least at two points $(u^*\theta^{*1})\ (u^*, \theta^{*2})$.

In our terms, if (u_i^*, Θ_i^*) is a solution set, then Θ_i^* contains at least two distinct values θ_i^{*j}.

Definition A set (u^*, Θ^*) is called a saddle set of $J(u,\Theta)$ if it satisfies

(a) $J(u,\theta^*(u)) \geq J(u^*,\theta^*) \geq J(u^*,\theta)$ (24a)

$\forall$ u, $\forall\ \theta\ \varepsilon\ \Theta$ and for some $\theta^*(u)\ \varepsilon\ \Theta^*$

(b) $J(u^*,\theta^*) = J \;\; \forall \; \theta^* \varepsilon \; \Theta^*$ (24b)

It is noted that the saddle set is a generalization of the saddle point. (A saddle point satisfies (a) for all $\theta^* \varepsilon \Theta^*$.)

Theorem 2 If assumptions A.1 and A.2 are satisfied, then $J(u,\theta)$ has a saddle set, (u^*, Θ^*) that satisfies

$$J(u^*, \theta^*(u)) \le J(u, \theta^*(u)) \tag{25}$$

$$\forall \; u \text{ and for some } \theta^*(u) \; \varepsilon \; \Theta^* .$$

An example of a function $J(u, \theta)$ with no saddle point but with a saddle set $\{(u_1^* \, \theta_i^*), \; i = 1, 2$ is given in figure 2. The proof of the theorem is given in [13]. The following corollary follows immediately from Theorem 2.

Corollary 1 Let $\bar{\theta}(u) \; \varepsilon \; \Theta$ be a point that satisfies:

$$\max_{\theta \varepsilon \Theta} J(u,\theta) = J(u, \bar{\theta}(u)) \tag{26}$$

and let $R(\tilde{u},\tilde{\theta})$ be the set in $\mathcal{U}$ that satisfies:

$$R(u,\theta) \triangleq \{u: \; \langle u-\tilde{u}, \; -\nabla_u J(\tilde{u},\tilde{\theta})\rangle \ge o, \; u\varepsilon\mathcal{U}\} \tag{27}$$

where <x,y> denotes the inner produce of x and y and $\nabla_u J(\tilde{u},\tilde{\theta})$ being the gradiant of $J(\cdot,\cdot)$ at $(\tilde{u},\tilde{\theta})$. Then, under assumptions A.1 and A.2 u^* satisfies

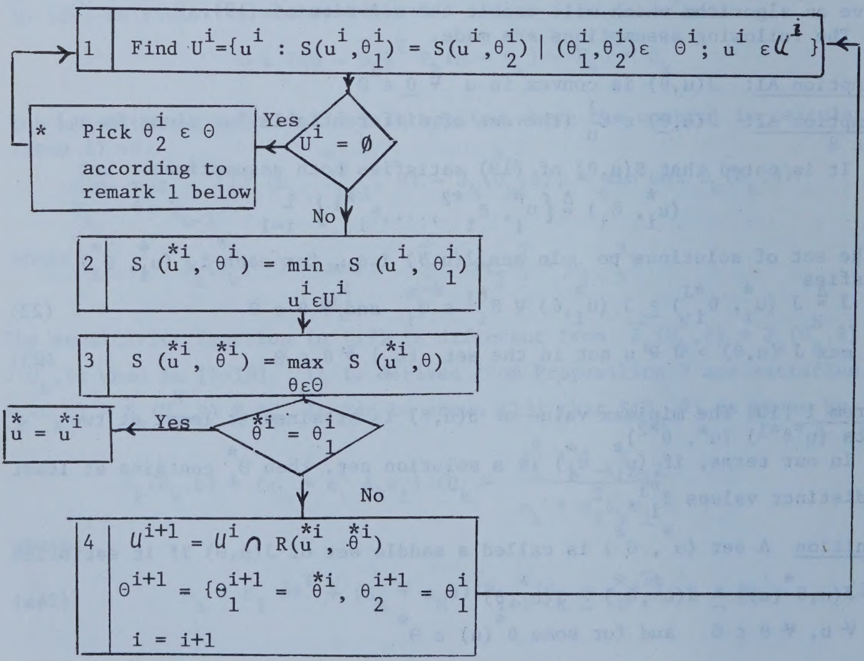
$$u^* \; \varepsilon \; R(\tilde{u},\tilde{\bar{\theta}}(\tilde{u})) \;\; \forall \; \tilde{u} \; \varepsilon \; \mathcal{U} \tag{28}$$

Consequently, the following algorithm for solving the minimax problem is introduced:

Algorithm 3

Initial values:

$\mathcal{U}^o = \mathcal{U}$; $\Theta^o = \{\theta_1^o, \theta_2^o\} \subset \Theta$, θ_1^o, θ_2^o arbitrary; i = o



1
Find $U^i = \{u^i : S(u^i,\theta_1^i) = S(u^i,\theta_2^i) \mid (\theta_1^i,\theta_2^i) \varepsilon \; \Theta^i; \; u^i \; \varepsilon \mathcal{U}^i\}$
$U^i = \emptyset$
Yes
* Pick $\theta_2^i \; \varepsilon \; \Theta$ according to remark 1 below
No
2
$S(u^{*i}, \theta^i) = \min_{u^i \varepsilon U^i} S(u^i, \theta_1^i)$
3
$S(u^{*i}, \theta^{*i}) = \max_{\theta\varepsilon\Theta} S(u^{*i},\theta)$
$\theta^{*i} = \theta_1^i$
Yes
$u^* = u^{*i}$
No
4
$\mathcal{U}^{i+1} = \mathcal{U}^i \cap R(u^{*i}, \theta^{*i})$
$\Theta^{i+1} = \{\theta_1^{i+1} = \theta^{*i}, \; \theta_2^{i+1} = \theta_1^i$
i = i+1

Remark 1. The choice of θ_2^i in the (*) step can be arbitrary however, choosing θ_2^i that satisfies $R\,(\overset{*}{u}^i, \theta_2^i) \cap R\,(\overset{*}{u}^i, \overline{\theta}\,(u^i)) = \emptyset$ will yield faster convergence than that achieved by choosing θ_2^i arbitrarily.

Remark 2. Steps 1, 2 of the algorithm can be replaced by letting U_i^* be an arbitrary point in $\mathcal{U}^i$. A "center" point in $\mathcal{U}^i$ is an example of such a choice. It is noted that $\mathcal{U}^i \subset \mathcal{U}$ is a polygon. The choice is a "center" point in the polygon guarantees convergence of the algorithm in the rate of approximately 2^{-n} after a closed polygon $\mathcal{U}^j$ has been established.

Remark 3. It is noted that the algorithm does not require the analytic solution of the maximation step. Furthermore, except for step 3, all steps are straightforward. Step 3 can be accomplished using some search technique.

Remark 4. The above algorithm has been applied [13] successfully to some examples in which other search techniques have been shown to fail [24].

Remark 5. One may terminate the algorithm when the polygon $\mathcal{U}^i$ is "small" enough. A criterion of smallness may be $\max\limits_{x,\,y \in \mathcal{U}^i} (\|x-y\|) \le \varepsilon$, where ε is some constant and $\|\cdot\|$ is some norm.

V. SIMULATION RESULTS

The adaptive control algorithm (algorithm 2) has been extensively simulated. Least square and the slower stochastic approximation algorithms have been employed in the identification step with the use of a proper bound on the error [27,29] to estimate Θ_i. The performance of our procedure was compared to that obtained using the certainty equivalence (C.E.) approach [17].

Typical plots of the cost, the output and the parameter estimates as functions of time are given in figures 3 and 5. We present on each plot the time history for the following four cases:

(a) Known parameters and no noise.

(b) Known parameters with noise.

(c) Unknown parameters with noise - C.E. approach.

(d) Unknown parameters - our approach.

The significant improvement over the C.E. approach using our approach is vivid. Similar results to those of figures 3-6 are reported in [13].

Computational wise, the most difficult step in Algorith 2 is that of solving the minimax problem of (19). Although Algorithm 3 provides a simple and relatively fast solution to this problem, still it may be too slow for practicle application of the control Algorithm 2. However, by choosing θ_1^o θ_2^o as two points on the boundaries of Θ such that $\|\theta_1^o - \theta_2^o\|$ is as large as possible, and solving only step 1 of Algorithm 2 usually yields a solution very close to the "optimal" control. Thus, a significant computational time in the solution of the minimax problem is saved to yield a very efficient (time wise and performance wise) control algorithm. It is important to note that the optimal control obtained by Algorithm 2 is usually different from the C.E. control obtained by assuming at each time instant k, that θ_k is the actual value of .

VI. CONCLUSIONS

An algorithm to solve optimal control problems in face of uncertainties has been developed from a game theoretic point of view. The algorithm has been derived so as to overcome the difficulties associated with the formulation of the control problem as a game of perfect information.

The algorithm is then specialized to the adaptive control problem where the system's parameters are constants but unknown. The adaptive control procedure is thus given in Algorithm 2. A major difficulty in Algorithm 2 is the solution of a minimax problem for a function which cannot be expressed explicitely in terms of the uncertainties. To facilitate the solution of the minimax problem the concept of a saddle set is introduced. This concept enables the solution of the minimax problem as in Algorithm 3.

Although the control procedure exhibits a significant improvement over the C.E. controller, still our solution is not the optimal one. To achieve optimality it is necessary to incorporate the dependence of the future uncertainty membership-sets on the present control and thus to achieve the closed-loop optimality. We have not expressed this dependence explicitely in our analysis and thus our control does not enhance learning as depicted from figure 5. Incorporation of the membership set dependence on the control is a subject for further investigation.

REFERENCES

[1] Aoki, M., Optimization of Stochastic Systems, Academic Press, New York, 1967.

[2] Hartwig, R.C. and Liberty, S.R., "A New Class of Statistical Performance Criteria for Stochastic Linear Control Systems", IEEE Aut. Cont., Vol. AC-16, No. 6, 1971.

[3] Tse, E., "On Optimal Control of Stochastic Linear Systems", IEEE Aut. Cont., Vol. AC-16, No. 6, 1971.

[4] Spang, A., "Optimum Control of an Unknown Linear Plant Using Bayesian Estimation of the Error", IEEE Trans. Aut. Cont., Vol. AC-10, 1965.

[5] Feldbaum, A.A., Optimal Control Systems, Academic Press, New York, 1965.

[6] Bertsekas, D.P. and Rhodes, I.B., "Sufficient Information Functions and The Minimax Control of Uncertain Dynamic Systems", IEEE Trans. Aut. Cont., Vol. AC-18, No. 2, 1973.

[7] Blaguire, A., Gerard F. and Leitmann, G., Quantitative and Qualitative Games, Academic Press, New York, 1969.

[8] Issacs, R. Differential Games, Wiley, New York, 1965.

[9] Rohrer, R.A. and Sobral, M. Jr., "Sensitivity Considerations in Optimal System Design", IEEE Trans. Aut. Cont., Vol. AC-10, 1965.

[10] Salmon, D.M., "Minimax Control Design", IEEE Trans. Aut. Cont., Vol. AC-13, No. 4, 1968.

[11] Darota, P. and Kestenbaum A., "Application of Game Theory to Sensitivity Design of Optimal Systems", IEEE Trans. Aut. Cont., Vol. AC-12, 1967.

[12] Ho, Y.C. and Sun, F.K., "On the Minimax Principle and Zero Sum Stochastic Differential Games", in Multicriteria Decision Making and Differential Games, Ed. G. Leitmann, Plenum Press, New York, 1967.

[13] Fogel, E., "Identification and Adaptive Control via Identification", Ph.D. Dissertation, Colorado State University, 1977.

[14] Ho, Y.C. and Sun, F.K., "Value of Information in Two-Team Zero Sum Problems", in Multicriteria Decision Making and Differential Games, Ed. G. Leitmann, Pelnum Press, New York, 1976.

[15] Feldbaum, A.A., "Theory of Dual Control", Automation and Remote Cont., Vol. 21, No. 9, No. 10, Vol. 22, No. 1, No. 2.

[16] Bellman, R., Dynamic Programming, Princeton, New Jersey, Princeton Univ. Press, 1957.

[17] Bar Shalom, Y. and Tse, E., "Concepts and Methods in Stochastic Control", in Control and Dynamic Systems, C.T. Leondes, Ed., Academic Press, 1975.

[18] Meier, L. III, Larson, R.E. and Ther, A.J., "Dynamic Programming for Stochastic Control of Discrete Systems", IEEE Trans. Aut. Cont., Vol. AC-16, No. 6, 1971.

[19] Leitman, G., "Cooperative and Non-Cooperative Many Plager Differential Games", Springer Verlag, N.Y., 1974.

[20] Witsenbauen, H.S., "A MInimax Problem for sampled Linear Systems", IEEE Trans. Aut. Cont., Vol. AC-13, 1968.

[21] Chang, S.L.H., and Peng, T.K.C., "Adaptive Guaranteed Cost Control of Systems with Uncertain Parameters", IEEE Trans. Aut. Cont., Vol. AC-17, No. 4, 1972.

[22] Speyer, T.L. and Shaked, U., "Minimax Design of a Class of Linear Quadratic Problems with Uncertain Parameters", IEEE Trsns. Aut. Cont., Vol. AC-19, 1974.

[23] Pearson, T.O., "Worst Case Design Subject to Linear Parameters Uncertainty" IEEE Trans. Aut. Cont., Vol. AC-19, 1971.

[24] Dem'yanov, V.F. and Malozemov, V.N., Introduction to Minimax, John Wiley and Sons, 1974.

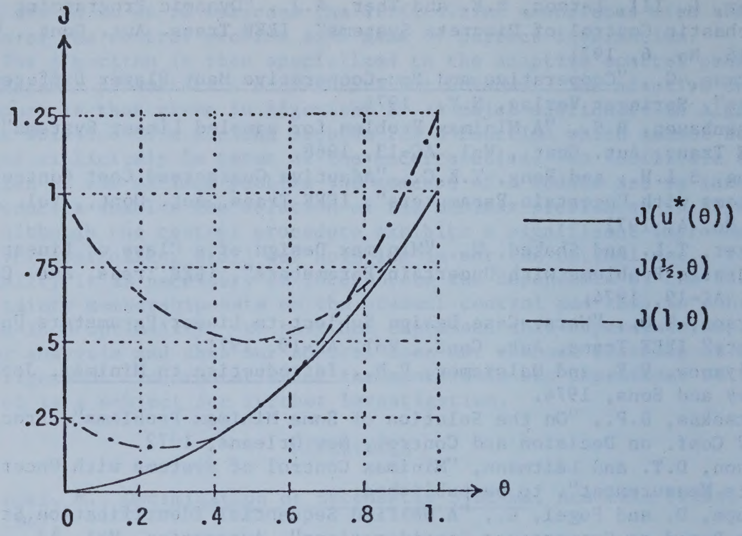
[25] Bertsekas, D.P., "On the Solution of Some Minimax Problems", Proc. 1972 Conf. on Decision and Control, New Orleans, 1972.

[26] Wilson, D.T. and Leitmann, "Minimax Control of Systems with Uncertain State Measurement", to be published.

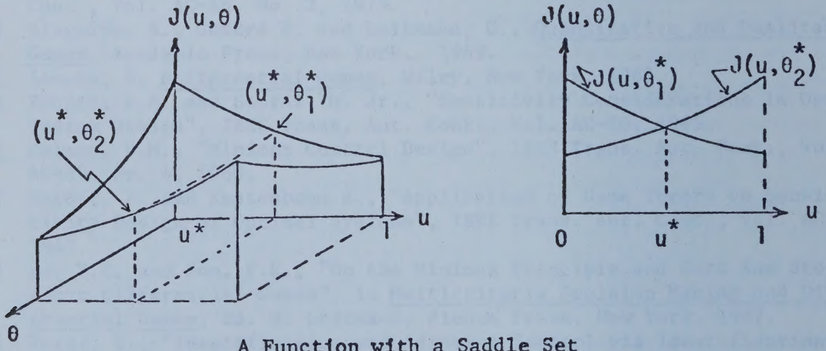
[27] Graupe, D. and Fogel, E., "A Unified Sequential Identification Structure Based on Convergence Considerations", Automatica, Vol. 12, pp. 53-59, 1976.

[28] Sage, A.P., Optimal System Control, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1968.

[29] Fogel, E. "System Identification via Membership Set Constrained with Energy Constrained Noise," submitted for publication.



Optimal and Suboptimal Cost Functions
Figure 1


J(u,θ)
(u*,θ₁*)
(u*,θ₂*)
u*
1
u
θ
J(u,θ)
J(u,θ₁*)
J(u,θ₂*)
0
u*
1
u

A Function with a Saddle Set

Figure 2

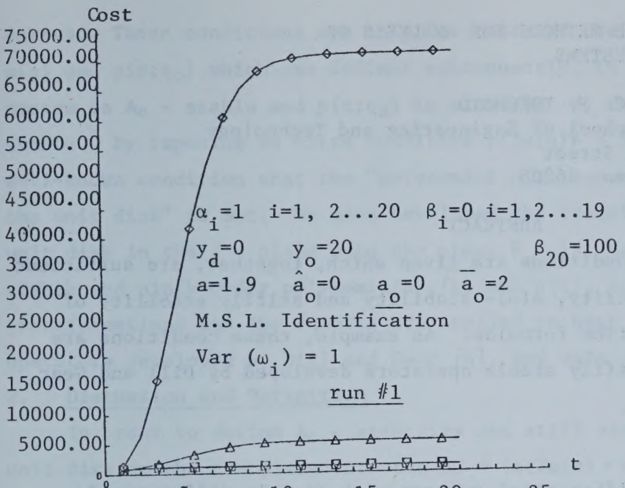


Figure 3.

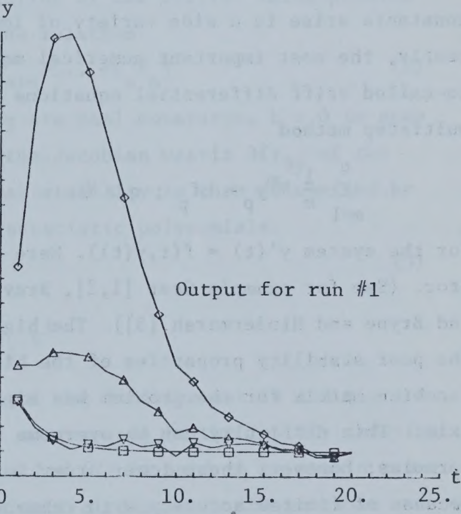


Figure 4

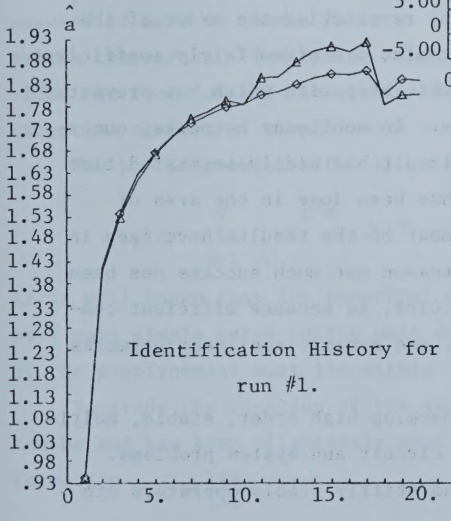


Figure 5

HIGH ORDER STIFFLY STABLE METHODS FOR ANALYSIS OF NONLINEAR CIRCUITS AND SYSTEMS

A. S. C. SINHA AND C. F. YOKOMOTO
Purdue University School of Engineering and Technology
IUPUI, 1201 E. 38th Street
Indianapolis, Indiana 46205

ABSTRACT

Two easy-to-check conditions are given which, together, are sufficient conditions for A_o - stability, A(o)- stability and stiffly stability of linear multistep integration formulae. As example, these conditions are applied to high order stiffly stable operators developed by Dill and Gear, and, Jain and Srivastava.

1. Introduction

Nonlinear ordinary differential systems with greatly different time constants arise in a wide variety of important physical problems. Currently, the most important numerical methods for solving such systems of so-called stiff differential equations are based on the backward difference multistep method

$$\sum_{m=1}^{q} \frac{1}{m} \nabla^m y_p = hf_p \;\; ; \; q > 0 \qquad (1)$$

for the system y'(t) = f(t,y(t)). Here ∇ is the backward difference operator. (See for example Gear [1,2], Brayton et. al. [3], Hindermarsh [4] and Bryne and Hindermarsh [5]). The biggest drawback of these formulas is the poor stability properties of the higher order formulas when the Jacobian matrix for the problem has eigenvalues close to the imaginary axis. This difficulty can be overcome by restricting the order of the formulas; however, these lower order formulas are often fairly inefficient because of limited accuracy with reasonable step size which has prevented transient analysis of nonlinear networks. In nonlinear networks, numerical instability can exist whenever a circuit had widely separated time constants [17-26]. Although much work has been done in the area of computer-aided circuit design [11-16], most of the results have been in the area of linear circuits. The main reason not much success has been achieved in the design of nonlinear circuits, is because efficient computational operators for analyzing d. c. and dynamic nonlinear networks have not been available.

The objective of this paper is to develop high order, stable, multistep methods for highly stiff nonlinear circuit and system problems. Sufficient conditions for A_o - stable and stiffly stable operators are derived. These conditions are only for roots of the polynomials $r(z)$, $s(z)$ and $p(z;q_o)$ which are defined subsequently. It is shown that if the system is A_o - stable and $p(z;q_o)$ is stable at ∞, the operator is stiffly stable. By imposing an extra condition $p(j\omega;q) \neq o$ for all ω and q, the well-known condition that the "polynomial $p(\zeta)$ has only simple zeros on the unit disk" is met. We have developed the transformations to map the unit disk in the ζ - plane onto the plane $H_- = \{z: Re(z) < o\}$ for seventh, eighth and ninth order polynomials $\rho(\zeta)$ and $\sigma(\zeta)$. As an example, these transformations and the results are applied to high order stiffly stable operators developed by Dill and Gear [6], and Jain and Srivastava [7].

2. Discussion and Definition

In order to define A_o - stability and stiff stability, we map the unit disk in the ζ - plane onto the $H_- = \{z:Re(z) < 0\}$. The general implicit method for the numerical solution of the initial value problem $\dot{y} = f(t,y)$, $y(o) = y_o$ is defined by the relation

$$\alpha_q y_{n+q} + \ldots\ldots + \alpha_o y_n = h[\beta_q f_{n+q} + \ldots + \beta_o f_n] \tag{2}$$

with α_q; $\beta_q \neq 0$, α_i, β_i, $i = 0,1,\ldots,q$ are real constants, $h > 0$ is step size; $t_m = mh$ and $f_m = f(t_m,y_m)$. If the Jacobian matrix $\partial f/_{\partial y}$ of the system is slowly varying, the numerical stability is then determined by the location of the roots of the characteristic polynomials.

$$\rho(\zeta) - h\,\lambda\,\sigma\,(\zeta) = 0 \tag{3}$$

where

$$\rho(\zeta) = \sum_{j=0}^{q} \alpha_j \xi^j, \quad \sigma(\zeta) = \sum_{j=0}^{q} \beta_j \zeta^j$$

and λ_i, $i = 1, \ldots, q$ are eigenvalues of the Jacobian matrix and ρ,σ are assumed to have no common factors. $h\lambda$ corresponds to the rapidly decaying system components. The system of equation (2) is said to be stable provided the roots of (3) lie within the unit circle, with the possible exception at most of simple roots occuring on the unit circle.

The p - polynomial with (1) is given by (See [8])

$$p(\zeta) = \sum_{m=1}^{q} \frac{1}{m} \zeta^{q-m} (\zeta-1)^m \tag{4}$$

It is well known that for numerical stability: (i) the p-polynomial may have only simple zeros on the unit circle, and (ii) the rest of the zeros of the p-polynomial must lie within the closed unit disk.

To study the location of the zeros of (4), a bilinear transformation can be and has been effectively used. In particular, the bilinear transformation $\zeta = \frac{1+z}{1-z}$

transforms the interior of the unit circle on the ζ-plane cnto the open left half-plane in the z-plane. It facilitates the problem of testing a polynomial p(ζ) for roots within the unit circle.

Define

$$\begin{aligned} p(z;q) &= (1-z)^q p(\zeta) \;/\; 2z \\ &= (1-z)^q p\left(\frac{1+z}{1-z}\right) \;/\; 2z \\ &= \sum_{m=1}^{q} \frac{1}{m}(1+z)^{q-m}(2z)^{m-1} \\ &= \sum_{i=0}^{q-1} a_i(q) z^{q-1-i} \end{aligned} \tag{5}$$

where

$$a_i(q) = \sum_{m=1}^{q-i} \frac{2^{m-1}}{m} \binom{q-m}{i}, \quad 0 \leq i \leq q-1$$

and $\binom{q-m}{i}$ is binomial coefficient

Definition 1

It is known that (1) is stable in the sense of Dahlquist if and only if (1) satisfies the root conditions: (i) the zeros of p lie inside or on the unit disk in the complex ζ-plane, and (ii) p has only simple zeros on the unit disk.

Definition 2 (Hurwitz Polynomial)

The real polynomials $p(z;q) = \sum_{i=o}^{q-1} a_i(q) \; z^{q-1-i}$, whose zeros all lie in the left half-plane are called Hurwitz Polynomials.

Define

$$\begin{aligned} r(z) &= (z-1)^q \rho\left(\frac{z+1}{z-1}\right) \\ &= \sum_{j=0}^{q} a_j z^j \end{aligned} \tag{6}$$

$$\text{where } a_{q-i} = \sum_{j=0}^{q} \alpha_{q-j} \left[\binom{q-j}{i} - \binom{j}{1}\binom{q-j}{i-1} + \binom{j}{2}\binom{q-j}{i-2} - \ldots + (-1)^{i-1}\binom{j}{i-1}\binom{q-j}{1} + (-1)^i\binom{j}{i}\right]$$

$$\begin{aligned} \text{and} \quad s(z) &= (z-1)^q \sigma\left(\frac{z+1}{z-1}\right) \\ &= \sum_{j=0}^{q} b_j z^j \end{aligned} \tag{7}$$

where

$$b_{q-i} = \sum_{j=0}^{q} \beta_{q-j}\left[\binom{q-j}{i} - \binom{j}{1}\binom{q-j}{i-1} + \binom{j}{2}\binom{q-j}{i-2}\right.$$

$$\left. - \ldots + (-1)^{i-1}\binom{j}{i-1}\binom{q-j}{1} + (-1)^{i}\binom{j}{i}\right].$$

For details of the a_{q-i} and the b_{q-i} [See 9]. Here $a_q = \rho(1) = 0$, and $b_q = \sigma(1) = 1$,

Define

$$p(z,q_o) = r(z) - q_o s(z), \quad q_o = \lambda h \varepsilon(-\infty,0]$$

$$= \sum_{j=0}^{q} (a_j - q_o b_j) z^j \; ; \qquad (8)$$

Definition 3 (A_o-stable)

The characteristic polynomial $\rho(\zeta) = h\lambda\ \sigma(\zeta)$ is A_o -stable if and only if the polynomial

$$p(z;q_o) = r(z) - q_o s(z)$$

is a Hurwitz Polynomial for all $q_o\ \varepsilon(-\infty,o]$

Definition 4 (A(o)-stable)

The characteristic polynomial $\rho(\zeta) = h\lambda\ \sigma(\zeta)$ is A(o)-stable if and only if the polynomial

$$p(z;q_o) = r(z) - q_o s(z)$$

is a Hurwitz Polynomial for all $q_o\ \varepsilon(-\infty,o)$

Definition 5 (Stiffly Stable)

The characteristic polynomial $\rho(\zeta) = h\lambda\ \sigma(\zeta)$ is said to be stiffly stable if it is convergent and if for some positive constants D, Θ and a we have

$$R_1 \cup R_2 \subset A \text{ and } R_3 \subset R$$

where sets

$$R_1 = \{\lambda \varepsilon\ C \mid \text{Re}\ \lambda < -D\}$$

$$R_2 = \{\lambda \varepsilon\ C \mid \text{Re}\ \lambda \leq -a,\ |\text{Im}\lambda| < \theta\}$$

$$R_3 = \{\lambda \varepsilon\ C \mid |\text{Re}\ \lambda < a \quad |\text{Im}\lambda| < \theta\}$$

$$A = \{\lambda \varepsilon\ C \mid |\zeta_i(\lambda)| < 1,\ i=1,..,q\}$$

$$R = \{\lambda \varepsilon\ \Omega \mid |\zeta_i(\lambda)| < |\zeta_1(\lambda)|,\ i = 2,3,..q\}$$

where $\Omega \subset C$ is called star region if $c\ \varepsilon\ \Omega$ implying $tc\ \varepsilon\ \Omega\ \forall\ t\ \varepsilon[o,1]$.

3. Theorems and Proof

Theorem 1. The backward difference multistep method (2),

$$\sum_{m=1}^{q} \frac{1}{m} \nabla^m y_p = hf_p$$

for the numerical integration of y'(t) = f(t,y(t)), is stable in the sense of Dahlquist if in Eq. (5)

(i) $\mathrm{Re}[p(z;q)] \leq \lambda \leq 0$ for $|z| < \mu(q)$

(ii) $p(j\omega;q) \neq 0$, for all ω; and q.

where $\mu(q) = \max\,[1, q\,\bar{a}(q)]$, $\bar{a}(q) = \max\limits_{1 \leq i \leq q-1} \left| \frac{a_i(q)}{a_o(q)} \right|$

Proof

We shall show that for $|z| \geq \mu(q)$. $\mathrm{Re}[p(z;q)] > 0$, the polynomial possesses no roots. In expanded form of (5)

$$p(z;q) = a_o(q) z^{q-1} + a_1(q)\, z^{q-2} + \ldots + a_{q-1}(q) = 0$$

so that

$$|p(z;q)| = |a_o(q)|\,|z|^{q-1} \left| 1 + \frac{a_1(q)}{a_o(q)} \frac{1}{z} + \ldots + \frac{a_{q-1}(q)}{a_{o}(q)} \right|$$

$$\geq |a_o(a)|\,|z|^{q-1} \Big| \Big[1 - \left| \frac{a_1(a)}{a_o(q)} \right| \frac{1}{|z|} - \left| \frac{a_2(q)}{a_o(q)} \right| \frac{1}{|z|^2} - \ldots - \left| \frac{a_{o-1}(q)}{a_o(q)} \right| \frac{1}{|z|^{q-1}} \Big] \Big|$$

$$\geq a_0(q)\,|z|^{q-1} \left[1 - \frac{(q-1)\bar{a}(q)}{q\bar{a}(q)} \right]$$

$$> 0$$

The only possibility for the characteristic roots to fall within Re z > 0 is if the variable z runs along z-plane from $-\mu(q)$ to $+\mu(q)$ on the imaginary axis. To avoid this, let $z = \sigma + j\omega$; $\sigma \geq 0$, $j\omega \neq 0$.
On the imaginary axis if $p(j\omega;q) \neq 0$ for all ω and q then $p(j\omega;q) \neq 0$ will not allow the roots to fall on the imaginary axis of the z-plane and therefore the characteristic roots must lie within the semi-plane $|z| < \mu(q)$ since we require that $\mathrm{Re}[p(z;q)] \leq 0$. The unit circle in the ζ-plane is mapped onto the half-plane Re $z \leq 0$ by the transformation $\zeta = \frac{z+1}{z-1}$, which implies the numerical stability in the sense of Dahlquist.

Theorem 2. The method $\rho(E)\, y_n = q_o\, \sigma(E)\, y'_n$, $(E\,y_n = y_{n+1})$ for the numerical integration of y'(t) = f(t,y(t)) is A_o-stable if

(i) $\mathrm{Re}[p(z;q_o)] \leq \lambda \leq 0$, $q_o \,\varepsilon\, (-\infty, o)$ for $|z| < \mu(q_o)$

(ii) $p(j\omega;q_o) \neq 0$ for all $\omega \neq 0$, and $q \geq 1$.

where $\mu(q_o) = \max\,[1, q\,\bar{a}(q_o)]$, $\bar{a}(q_o) \equiv \max\limits_{1 \leq i \leq q-1} \left| \frac{(a_i - q_o b_i)}{(a_o - q_o b_o)} \right|$,

and $p(z;q_o) = r(z) - q_o\, s(z)$, $q_o = \lambda h \,\varepsilon\, (-\infty, 0)$.
Furthermore, if $\mathrm{Re}[\rho(z;q_o)] \leq \lambda \leq 0$; $q_o \,\varepsilon\, (-\infty, 0]$ for $|z| < \mu(q_o)$, the method

$\rho(E)\, y_n = q_o\, \sigma(E)\, y_n'$ is A(o) stable.

Remark 1. The assumptions of Theorem 2 also satisfy all conditions for the stiffly stable method of Jeltsch [10] and if $p(z;q_o)$ is stable at ∞, we conclude the method is stiffly stable.

Remark 2. The proof of Theorem 2 follows on arguments similar to those applied to Theorem 1.

4. Examples

Example 1: (Dill and Gear, [6])

From (6), coefficients for $r(z) = \sum_{j=0}^{q} a_j z^j$, and $\rho(\zeta) = \overset{q}{\underset{j=0}{\zeta}} a_j \zeta^j$ are related for q = 7 by a linear transformation given by

$$\begin{bmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 21 & 9 & 1 & -3 & -3 & 1 & 9 & 21 \\ -35 & -5 & 5 & 3 & -3 & -5 & 5 & 35 \\ 35 & -5 & -5 & 3 & 3 & -5 & -5 & 35 \\ -21 & 9 & -1 & -3 & 3 & 1 & -9 & 21 \\ 7 & -5 & 3 & -1 & -1 & 3 & -5 & 7 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \end{bmatrix}$$

$s(z)$ and $\sigma(\zeta)$ polynomials are also related by the same transformation. The transformation T can be checked by a laborious computation. Next, we take the stiffly stable example of Dill and Gear [6] of order seven. The seventh-order method

$$\sigma(\zeta) = \zeta^7 + 0.7\zeta^4$$

$$\rho(\zeta) = -2.56\zeta^7 + 7.07\zeta^6 - 10.920\zeta^5 + 11.492\zeta^4 - 8.050\zeta^3 + 3.99\zeta^2 - 1.12\zeta + .138$$

is stiffly stable. Let

$$s(z) = (z-1)^7\, \sigma\left(\frac{z+1}{z-1}\right)$$

$$= 1.7z^7 + 7.7z^6 + 18.9z^5 + 32.9z^4 + 37.1z^3 + 23.1z^2 + 6.3z + 0.3 \text{ and}$$

$$r(z) = (z-1)^7\, \rho\left(\frac{z+1}{z-1}\right)$$

$$= -[3.124z^6 + 14.568z^5 + 37.556z^4 + 69.544z^3 + 86.652z^2 + 70.936z + 45.34]$$

Routh-Hurwitz criterion shows that $s(z)$, $-r(z)$ and $p(z)$ are Hurwitz Polynomials, so that Re $p(z) < 0$, is stiffly stable since the constants a, D and Θ exist.

Example 2. (Dill and Gear, Ref. 6)

The eight-order method

$$\sigma(\zeta) = \zeta^8 - 1.8\zeta^7 + 0.81\zeta^6$$

$$\rho(\zeta) = -0.162 + 1.489\zeta - 6.129\zeta^2 + 14.89\zeta^3 - 23.763\zeta^4 + 26.587\zeta^5 - 21.070\zeta^6 + 10.636\zeta^7 - 2.478\zeta^8$$

is stiffly stable. To show this, we first map from $\zeta \leftrightarrow z$ the polynomials

$r(z) = (z-1)^8 \rho(\frac{z+1}{z-1}) = \zeta_{j=0}^{8} a_j z^j$ and $s(z) = (z-1)^8 \sigma(\frac{z+1}{z-1}) = \zeta_{j=0}^{8} b_j z^j$

are found to be related by a transformation

$$\begin{bmatrix} a_8 \\ a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -8 & -6 & -4 & -2 & 0 & 2 & 4 & 6 & 8 \\ 28 & 14 & 4 & -2 & -4 & -2 & 4 & 14 & 28 \\ -56 & -14 & 4 & 6 & 0 & -6 & -4 & 14 & 56 \\ 70 & 0 & -10 & 0 & 6 & 0 & -10 & 0 & 70 \\ -56 & 14 & 4 & -6 & 0 & 6 & -4 & -14 & 56 \\ 28 & -14 & 4 & 2 & -4 & 2 & 4 & -14 & 28 \\ -8 & 6 & -4 & 2 & 0 & -2 & 4 & -6 & 8 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{bmatrix}$$

or $\bar{a} = T\,\bar{\alpha}$

$s(z)$ and $\sigma(\zeta)$ polynomials are also related by the transformation $\bar{b} = T\bar{\beta}$, where T is as defined above. The transformation T can be checked by a Laborious Computation. We find that

$$r(z) = -[0.016z^7+0.868z^6+12.056z^5+55.388z^4+127.808z^3+174.46z^2 +156.568z+107.204]$$ and

$$s(z) = [0.01z^8+0.44z^7+6.04z^6+27.56z^5+61.9z^4+77.96z^3+56.44z^2 +22.04z+3.61]$$

Routh-Hurwitz criterion shows that $s(z)$, $-r(z)$ and $p(z)$ are Hurwitz polynomials, so that $\mathrm{Re}[p(z)] < 0$, and all conditions of Theorem 2 hold. The method is stiffly stable since it is A(0)-stable and the constants a, D and Θ exist [see Dill and Gear, Ref. 6]

Example 3. [Jain and Srivastava, Ref.7]

The ninth-order method

$$\sigma(\zeta) = \zeta^9-2.7\zeta^8+2.43\zeta^7-0.729\zeta^6$$

$$\rho(\zeta) = 0.1597-1.6275\zeta+7.5069\zeta^2-20.7005\zeta^3+38.0317\zeta^4-49.4460\zeta^5 +46.7195\zeta^6-31.1432\zeta^7+12.9916\zeta^8-2.4923\zeta^9$$

is stiffly stable. To show this, we again map from ζ-plane to z-plane the polynomials

$r(z) = (z-1)^9 \rho(\frac{z+1}{z-1}) = \sum_{j=0}^{9} a_j z^j$ and $s(z) = (z-1)^9 \sigma(\frac{z+1}{z-1}) = \sum_{j=0}^{9} b_j z^j$

are found to be related by a transformation

$$\begin{bmatrix} a_9 \\ a_8 \\ a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_o \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -9 & -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 & 9 \\ 36 & 20 & 8 & 0 & -4 & -4 & 0 & 8 & 20 & 36 \\ -84 & -28 & 0 & 8 & 4 & -4 & -8 & 0 & 28 & 84 \\ 126 & 14 & -14 & -6 & 6 & 6 & -6 & -14 & 14 & 126 \\ -126 & 14 & 14 & -6 & -6 & 6 & 6 & -14 & -14 & 126 \\ 84 & -28 & 0 & 8 & -4 & -4 & 8 & 0 & -28 & 84 \\ -36 & 20 & -8 & 0 & 4 & -4 & 0 & 8 & -20 & 36 \\ 9 & -7 & 5 & -3 & 1 & 1 & -3 & 5 & -7 & 9 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_o \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \\ \alpha_9 \end{bmatrix}$$

or, $\overline{a} = T\ \overline{\alpha}$

$s(z)$ and $\sigma(\zeta)$ polynomials are also related by the transformation $\overline{b} = T\ \overline{\beta}$, where T is as defined above. Then

$$r(z) = -\ [.0025z^8+0.1248z^7+2.8824z^6+28.5018z^5+118.0642z^4+260.324z^3+ 347.144z^2+308.1949z+210.8189] \text{ and}$$

$$s(z) = [0.001z^9+0.063z^8+1.44z^7+14.232z^6+58.554z^5+125.406z^4+153.768z^3 + 109.44z^2+42.237z+6.859$$

It can be easily verified that $s(z)$, $-r(s)$, and $p(z)$ are all Hurwitz Polynomials, so that all conditions of Theorem 2 hold. Thus, the method is stiffly stable since it is A(0)-stable and the constants a, D, and Θ exist. (See [7]).

Acknowledgement

I am thankful to Dr. R. J. Lambert, Senior Professor of Computer Science and Mathematics of Ames Research Lab., ERDA, Iowa State University for his valuable comments and discussions during the course of his research. The author was a visiting research professor at Ames Research Lab., ERDA during the Summer of 1977. The author also wishes to acknowledge his gratitude to Mr. Paul Brink for editing this paper.

References

1. C.W. Gear, Algorithm 407: DIFSUB for solution of ordinary differential equations, comm. ACM 14, 3 (March 1971) 176-179.

2. C.W. Gear, Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, N.J., 1971.

3. R.K. Brayton, F.C. Gustaron; and Hachtel, G.D., A new efficient algorithm for solving differential algebraic systems using implicit backward differentiation formulas. Proc. IEEE 60 (1972), 98-108.

4. A.C. Hindermarsh, Ordinary differential equation solver. UCID-3001, Rev. 3, Lawrence Livermore Laboratory, U. of California, Livermore, 1974.

5. G.D. Bryne, and A.C. Hindermarsh, A polyalgorithm for the numerical solution of ordinary differential education. ACM Trans, Math Software, 1 (March 1975), 71-96.

6. C. Dill and C.W. Gear, A Graphical Search for Stiffly Stable Methods for Ordinary Differential Equations, J. Assoc. Comput. Mach.,18, 1971, pp. 75-79.

7. M.K. Jain and V.K.Srivastava, High Order Stiffly Stable Methods for Ordinary Differential Equations., Rep. No. 394, Dept. of Computer Sci., University of Illinois, Urbana, Ill., April, 1970.

8. C.W. Cryer, A New Class of Highly-Stable Methods: A_o-Stable Methods, BIT, 13, (1973, pp. 153-159).

9. E.I. Jury, Theory and Application of the Z-Transform Method, New York, Wiley, 1964, p. 83.

10. R. Jeltsch, Stiff Stability and Its Relation to A_o- and A(o) - Stability, SIAM J. Numer. Anal., Vol. 13, No. 1, 1976,pp.8-17.

11. D.A. Calahan, Computer Aided Network Design, New York, N.Y., McGraw-Hill Book Company, 1968.

12. F.F. Kuo and W.G. Magnuson, Jr., Computer-Oriented Circuit Design, Englewood Cliffs, N.J., Prentice-Hall, Inc. 1969.

13. S.W. Director and R.A. Rohrer, "Automated Network Design--The Frequency Domain Case," IEEE Transactions on Circuit Theory.

14. S.W. Director and R.A. Rohrer, "On the Design of Resistance N-Port Networks by Digital Computer," IEEE Transactions on Circuit Theory, Vol. CT-16, pp. 337-346, August 1969.

15. F.H. Branin and H.H. Wang, "A Fast Reliable Iteration Method for dc Analysis of Nonlinear Networks," Proc. IEEE, Vol. 55 pp. 1819-1826, November, 1967.

16. H. Schiechman, "Computation of DC Solutions for Bipolar Transistor Networks," IEEE Transactions on Circuit Theory, Vol. CT-16, pp. 460-466, November, 1969.

17. I.W. Sandberg and H. Schichman, "Numerical Integration of Stiff Nonlinear Differential Equations," Bell Sys. Tech. J., Vol. 47, pp. 511-527, April, 1968.

18. D.A. Calahan, "A Stable Accurate Method of Numerical Integration for Nonlinear Systems, "Proc. IEEE, Vol. 56, No. 4, p. 744, April, 1968.

19. T.W. Stern, "Computer-Aided Analysis of Nonlinear Networks Described by Constrained Differential Equations," Proceedings of the International Symposium on Network Theory," University of Belgrade, Yugoslavia,

September, 1968, pp. 178-190.

20. R.A. Allen and C. Pottle, "Stable Integration Methods for Electronic Circuit Analysis with Widely Separated Time Constants," Proc. Sixth Allerton Conference on Circuit and System Theory, Urbana, Illinois, pp. 311-320, October, 1968.

21. D.A. Calahan, "Numerical Considerations in the Transient Analysis and Optimal Design of Nonlinear Circuits,: Proceedings of SIAM Symposium, pp. 27-58.

22. F.G. Gustavson, et. al., "Symbolic Generation of an Optimal Crout Algorithm for Sparse Systems of Linear Equations," Proceedings Sparse Matrix Symposium, IBM Watson Research Center, Yorktown Heights, New York, pp. 1-10, September, 1968.

23. G.D. Hachtel, R. Brayton, and F. Gustavson, "A Sparse Matrix Approach to Network Analysis," Proceedings Cornell Conference on Computerized Electronics, August, 1969.

24. D.A. Calahan, "Switching Circuit Optimization," Prox. Cornell Conference on Computerized Electronics, August, 1969.

25. C.W. Ho, and G.D. Hachtel, "Automated Optimization of Switching Networks via SCEPTRE simulation," to appear in the IEEE Transactions on Circuit Theory.

26. L.O. Chua, Efficient Algorithms for Implementing the Iterative Piecewise-Linear Method, Purdue University Report TR-EE-69-21, September, 1969.

A FAST, SPACE-EFFICIENT AVERAGE-CASE ALGORITHM FOR THE 'GREEDY' TRIANGULATION OF A POINT SET, AND A PROOF THAT THE GREEDY TRIANGULATION IS NOT APPROXIMATELY OPTIMAL

GLENN K. MANACHER
Dept. of Information Engineering and Computer Center
University of Illinois at Chicago Circle
Chicago, Illinois 60680

and

ALBERT L. ZOBRIST
Jet Propulsion Laboratory
Pasadena, California 91103

ABSTRACT

A Greedy Triangulation (GT) of a planar point set S is a triangulation T in which the $n(n-1)/2$ edges are considered in sequence, ordered from smallest to largest. Each edge considered is inserted into the plane provided no edge already in the plane intersects it. Assuming no pair of edges is of the same length, the GT is clearly unique.

It was formerly thought that the GT might be an approximation algorithm to the Optimum Triangulation (OT) which minimizes the sum of edge lengths; that is, that the GT might always be within a fixed factor of the OT. This has been disproved and a construction given in which the GT is worse than the OT by a factor of order $\sqrt[3]{n}$. Lloyd [1] has shown that the GT is not the OT and conjectured that finding the OT may be NP-complete. We mention here that the problem is open whether there exists an efficient approximation algorithm to the OT.

The problem addressed in this paper is how to find the GT efficiently in the average case. We show first how in the worst case, the GT may be obtained in time $O(n^3)$ and space $O(n)$, then how the algorithm may be slightly modified to produce a time $O(n^2)$, space $O(n)$ solution in the average case. We mention here that Gilbert [2] has found a worst-case solution using totally different techniques that requires space $O(n^2)$ and time $O(n^2 \log n)$.

Any triangulation of $n = |S|$ points may be shown easily to contain exactly

$$\#(T) = 3n - 3 - C(S) \tag{1}$$

edges, where $C(S)$ is the number of points on the convex hull of S.

From (1) it follows immediately that there exists an algorithm for computing the GT in time $O(n^3)$ and space $O(n^2)$.

ALGORITHM 1. NAIVE GT ALGORITHM

1. Obtain the $n(n-1)/2$ edges. Sort them according to length. Set $T \leftarrow \phi$.
2. Consider each edge e in sequence, from smallest to largest. For each e, consider all $t \in T$. If no t crosses e, place e in T.

The reader may use (1) to prove the time and space assertions easily.

1. The worst case. We now show how to reduce the space requirement to $O(n)$. We first introduce an array RAW(1:n), containing the points in S in arbitrary order. If $s \in S$ and $s = \text{RAW}[i]$, we shall define $i = \text{index}(s)$.

It proves convenient to deal with directed edges ("d-edges") defined in such a way that for every edge e, there are two d-edges, one "proper" and the other "improper." A proper (d-)edge pq, $p,q \in S$, $p \neq q$, is a d-edge such that index(p) < index (q). From here on, where not explicitly specified otherwise, we shall deal only with proper d-edges. We shall sometimes omit "proper" and "d-" when they are clear in context.

We now define a heap from which we shall repeatedly draw the smallest element. This heap is of a naive, linear type; it does not support any logarithmic operations. The heap, ML ("merge-list") contains up to n-1 d-edges at any time. Initially, for all $p \in S$, the heap contains the proper d-edge pq of minimum length, assuming one exists (as it will except for the point with index n). At each step of the algorithm, the smallest (length) edge yz is removed from the heap and compared with all $t \in T$; if not crossed, it is placed in T. A new proper edge yz′ that is of minimum length subject to the constraint $|yz'| \geq |yz|$ is then placed in the ML, if there is one.

We now formalize these notions, incorporating the case of equal-length edges. (The GT will still be unique for a given assignment of points to RAW.)

The first thing to do is to impose a total ordering on the edges. For two nonidentical edges pq, rs, we say that pq $<^T$ rs if either

(1) $|pq| < |rs|$, or
(2) $|pq| = |rs|$, $p \neq r$, and index(p) < index(r), or
(3) $|pq| = |rs|$, $p = r$, and index(q) < index(s). (2)

"$>^T$" will be the reflexive dual of "$<^T$." "Min" will be defined as "least over $<^T$."

Define now

$$\{p^* = pq:\ q \in S \text{ and } q \neq p\}. \quad (3)$$

Finally, it is natural to define a successor function over $>^T$ since $>^T$ is a total ordering. The function Next is defined so that it returns a singleton set if the successor exists, else the void set if it does not.

$$\text{Next}(pq) = \underline{\text{if}}\ \exists\ pt \text{ such that } pt >^T pq\ \underline{\text{then}}$$
$$\{(pr:\ pr >^T pq \text{ and } \nexists\ ps \text{ such that } pr >^T ps >^T pq)\}$$
$$\text{else } \phi. \quad (4)$$

In similar fashion, define Least(p) by

$$\text{Least}(p) = \underline{\text{if}}\ \exists\ pr \text{ such that } pr >^T pq\ \underline{\text{then}}$$
$$\{(pr:\ \nexists\ ps \text{ such that } ps >^T pr)\}\ \text{ else } \phi. \quad (5)$$

We now present our improved algorithm.

ALGORITHM 2. ALGORITHM USING O(n) SPACE

```
1.  (initialization).  ML ← $\bigcup_{p \in S}$ Least(p);  T ← $\phi$;
2.  (Iterative step).
    2a.  e ← Min(ML)  */ select minimum element of heap /*
    2b.  ML ← ML - {e}  */ remove from heap /*
    2c.  If $\nexists$ some $t \in T$ such that t crosses e, then T ← T $\cup$ {e}.
    2d.  ML ← ML $\cup$ Next(e)  */ place successor to e on heap, if
         there is one /*
    2e.  If ML = $\phi$ then stop else goto 2a.
```

Lemma 1. Algorithm 2 obtains a GT in time $O(n^3)$ and space O(n).

Proof. The time and space estimates are virtually self-evident: Each edge requires O(n) operations for finding, inserting, checking against the edges in T, etc. and there are $O(n^2)$ edges. Only the claim that it produces a GT needs attention. This will clearly be the case if claim 1 is true.

Claim 1. Step 2b successively picks every edge in E (the set of all edges) once, and picks them in nondecreasing order of length.

Proof. For all p, the "Next(e)" in step 2d will have the effect of accessing all the proper elements of p*, bringing them one by one to the heap, in increasing order. The heap mechanism guarantees that these will be released in sequence. Since proper d-edges are one-to-one with edges, the lemma follows. □

2. The average case. We use extentions of the methods of section 1 to improve the average case performance to $O(n^2)$. The basic idea is to introduce a "cheap" (O(1)) test that will determine for any edge that either (a) it is crossed by some edge in T or (b) to within some fixed (hopefully, large) probability, it is not crossed. The test is used in step 2d; only an edge that passes the cheap test is admitted to the ML. We then use a combination of geometrical and statistical arguments to show that the probability for such an edge to be still uncrossed by any $t \in T$ at the time it reaches the top of the heap is bounded below by some constant. Consequently, the number of edges reaching the top of the heap that are destined to enter T is proportional to $|T|$, and therefore to n, rather than to n^2, as in algorithm 2.

Our "average model" states simply that n points are distributed at random inside a circle of unit radius.

First some terminology. An edge that is selected by step 2a of algorithm will be said to be on top of the heap. An edge that has yet to be placed on the heap, but which is nevertheless examined, will be called a trial edge. Finally, an algorithm which, like algorithm 2, places edges in T in increasing order of $>^T$, and never removes an edge once placed in T, will be called conformal. This definition is not vacuous, since conceivably an algorithm could produce a GT without being conformal.

Let pq be an edge on top in step 2a. Suppose that pq is crossed by edge $rs \in T$. Then we have

Definition 1. rs will be called end-near p iff $|ps| < |pq|$ and $|pr| < |pq|$.

Lemma 2. At the time pq is on top, if $rs \in T$ crosses pq closer to p than any other edge in T crossing pq, if rs is end-near p, and if the algorithm is conformal,then $ps \in T$ and $pr \in T$.

Proof. Consider just pr; what we prove will work for ps by symmetry.

We know that pr must have already been on top, by conformality. If it is not in T, then at the time it was on top, it must have been blocked, i.e., crossed by some edge in T at that time. By conformality, the blocking edge will be in T at the time pq is on top. Let ab be the edge that blocked pr. Clearly ab cannot cross rs. If ab crosses pr and ps, then it must cross pq closer to p then o, violating the hypotheses. (These entities are all shown in figure 1 for easy visualization.) Thus, it must cross pr but not ps or rs. Hence one of its end-points must lie inside Δprs.

We now show that this is impossible. Let Z_{pro} denote the set of points in S inside triangle pro. Let z^o_{pro} denote the point in Z_{pro} such that angle $z^o_{pro}rs$ is minimum over all points in Z_{pro}. Similarly, let z^o_{pso} be the point in Z_{pso} such that angle $z^o_{pso}rs$ is minimum. Define z* to be either z^o_{pso} or z^o_{pso}. whichever is nearer to rs (or, if one does not exist, then the other). It may be shown rather easily that neither z*r nor z*s is blocked. Hence both z*r and z*s are in T, by conformality and because $|z^*r| < |pq|$ and $|z^*s| < |pq|$. One of these edges must cross pq closer to p than o, violating the hypothesis. Hence there are no points inside Δprs, therefore pr and ps were unblocked when they were on top, and so they are in T. □

<u>Lemma 3</u>. At the time pq is on top, suppose at least two nondirected edges with one vertex at p are in T. Suppose further that pr and ps, two edges in T, exist such that pr and ps "include" pq in the sense of figure 1, so that (a) $\alpha + \beta < 180^{\circ}$, and (b) α and β cannot be made smaller by choosing pr′ instead of pr or ps′ instead of ps. Then if $|rs| < |pq|$, it follows that rs $\in$ T.

<u>Proof</u>. By the same line of reasoning as the proof of lemma 2. If rs is blocked, then there must be a point in Δprs. Choose a point y such that angle ypr is minimum if the point is in Δpro, or a point x such that angle xps is minimum if the point is in Δpso. Then py and/or px is unblocked, hence in T, violating the assumptions about the minimality of α and β. □

These lemmas lead to an essential

<u>Definition 2</u>. An edge pq is <u>end-triangulated at p</u> iff there exists a triangle prs in T such that rs intersects pq. An edge pq is <u>end-triangulated</u> (ET) iff it is end-triangulated at p or at q.

<u>Definition 3</u>. A <u>simple triangle</u> is a triangle in T containing no points.

<u>Lemma 4</u>. Let pq be on top and be ET at p by Δprs, and suppose rs intersects pq closer to p than any other edge intersecting pq. Then (a) Δprs is simple, and (b) angle rpq and angle spq are smaller respectively than any angles r′pq and s′pq for all edges r′p and s′p in T.

<u>Proof</u>. The proof of (a) is essentially identical to the proof of lemma 2. (b) follows almost at once: Any purported r′p with the assumed property would either have to cross rs, or r′ would have to lie within Δ prs. But by (a) Δprs is simple. □

Our key lemma now follow.

<u>Lemma 5</u>. Let pq be on top. Suppose pq is intersected by one or more edges in T. Let rs be the edge intersecting pq whose point of intersection lies nearest p. To determine whether rs is end-near p, it is necessary and sufficient that algorithm 3 return TRUE.

ALGORITHM 3. END-NEARNESS ROUTINE EN(pq,p)

```
1.  Are there less than 2 non-directed edges with vertices at p?
        If so, return FALSE.
2a. Find point r so that pr $\in$ T and angle rpq is minimum;
2b. Find point s so that ps $\in$ T and angle spq is minimum, in
        opposite sense (viz. figure 2).
3.  If angle rpq + angle spq $> 180^{\circ}$, return FALSE.
4.  If $|\vec{r} - \vec{s}| < |pq|$ then return TRUE else return FALSE.
```

<u>Proof</u>. If the hypothesis is true, then by lemma 2, pr and ps will be in T. By lemma 3, it suffices to check pr and ps to make sure their angles "contain" pq and that $|rs| < |pq|$. By lemma 4, the angles rpq and spq may chosen to be minimum. □

<u>Corollary</u>. Algorithm 3 determines whether pq is end-triangulated by a simple triangle at point p.

<u>Lemma 6</u>. If pq is not intersected by any edge in T at the time it is on top, then algorithm 3 will return FALSE.

<u>Proof</u>. Otherwise lemma 3 would be false (see step 4 of algorithm 3).

<u>Lemma 7</u>. If algorithm 3 returns TRUE, then pq is crossed by some e $\in$ T.

<u>Proof</u>. Trivial.

<u>Lemma 8</u>. Algorithm 4 determines whether pq is ET by a simple triangle at either end, when pq is on top.

ALGORITHM 4. SIMPLE ET ROUTINE SEN(pq);

```
Return EN(pq,p) or EN(pq,q).
```

Proof. Obvious.

Lemma 9. If pq is on top and is crossed by some $t \in T$, then the probability c that SEN with return FALSE is less than .025.

Sketch of proof. If pq is crossed but SEN(pq) returns FALSE, then the edge crossing nearest p must not be end-near p, and the edge crossing nearest q must not be end-near q. Three cases may be discerned:

Case 1. pq is crossed once. In this case, the same crossing edge crosses nearest p and nearest q. It is easy to show that the crossing edge must be at least $\sqrt{3}/2$ as large as $|pq|$, and must cross pq within $|pq|(1-\sqrt{3}/2)$ of one of its ends. In addition, there are angular restraints, and it is possible to show on the basis of the distribution (7) that SEN(pq) will return FALSE in this case less than 1% of the time.

Case 2. pq is crossed twice. In this case, one crossing edge crosses closest to p and one crosses closest to q. Assuming both edges lie almost parallel to pq, thus somewhat increasing the probability that they will not be end-near, it is possible to show analytically that SEN(pq) will return FALSE less than 2.5% of the time.

Case 3. pq is crossed more than twice. This case cannot easily be analyzed, but it is easy to show that SEN(pq) will return FALSE less often than in case 2, since the crossing edges crossing nearest to p and q will be closer to them.

Hence case 2 establishes a "worst case" for all cases, which proves the lemma. □

Lemma 9.1. If SEN returns FALSE, then the probablity that pq is crossed is less than c {lemma 9} and hence less than .025.

Proof. By lemmas 9 and 7 and Bayes´ rule. □

We observe here that all of these lemmas have assumed that pq is on top. Our real wish is to use algorithm 4 to check other edges in p* that are "candidates for admission" to ML just following the removal of pq from the top. Let pq´ be one such edge. Clearly $pq' >^{T} pq$. Whether pq´ is ET at p may be tested by calling routine EN´(pq´,p,pq), which is identical to EN(pq,p) except that q´ is substituted for q in lines 2 and 3, but not in line 4. We then correspondingly substitute SEN´(pq´,pq) for SEN(pq); this computes EN´(pq´,p,pq) or EN´(pq´q,pq). The analogue to lemma 9.1 using SEN´(pq´,pq) will be the same but with a constant larger than .025, since the crossing lines will be shorter relative to pq´ than to pq. (See lemma 11.3.)

We now modify Next(pq) so that it selects the first successor to pq that is not ET. We call the routine "Nextut", for "next untriangulated."

Nextut(pq) = if $\exists$ pt such that $pt >^{T} pq$ and SEN´(pt,pq) = FALSE then {(pr: $pr >^{T} pq$, SEN´(pr,pq) = FALSE and $\nexists$ ps such that $pr >^{T} ps >^{T} pq$ and SEN´(ps,pq) = FALSE)} else ϕ (6)

We now claim that algorithm 5 will run in average time $O(n^2)$.

```
ALGORITHM 5.  FAST, SPACE-EFFICIENT GT ALGORITHM

1.  (initialization).  ML ← $\bigcup_{p \in S}$ Least(p);  T ← $\phi$ ;
2.  (Iterative step).
    2a.  e ← Min(ML)  */ select minimum element of heap /*
    2b.  ML ← ML - { e }*/ remove from heap /*
    2c.  If $\nexists$ some $t \in T$ such that t crosses e, then T ← T $\cup$ {e}.
    2d.  ML ← ML $\cup$ Nextut(e)  */ place successor to e on heap,
         if there is one /*
    2e.  If ML = $\phi$  then stop else goto 2a.
```

We first need a way to make algorithm 4 run in average time O(1). To do this, we introduce an auxiliary data structure consisting of an array U(1:n) whose i^{th} cell contains the header to a linear linked list of all undirected edges one of whose vertices is p, where i=index(p). In step 2 of algorithm 3, we simply scan U[index(p)].

Lemma 10. Using U, SEN(pq′,pq) may be computed in average time O(1).

Proof (sketch). There are on the average 6 edges at each vertex in a triangulation, with small variation. Hence the scans require average time O(1). □

Lemma 11. Let step 2d of algorithm 5 select some edge g=Nextut(pq) and place it on the ML. When g rises to the top of the ML, its probability for being uncrossed by any $t \in T$ will be lower-bounded by some constant $c > 0$, where c is independent of $|T|$ or of n.

Proof. Define sets NN(k) of directed edges as follows: $pq \in NN(k) \Leftrightarrow pq$ is the k^{th} least element of p* when p* is ordered by $<^1$. Hence sets NN(1), NN(2), ... NN(n-1) are defined, each containing n d-edges.

If $e \in NN(k)$, then the mean length of e, $\bar{e}$, may be closely approximated by assuming that one of its endpoints lies at the center of the unit circle enclosing S. Since the points have uniform density, it follows that k points lie in a circle satisfying $(\pi \bar{e}^2 / \pi 1^2) = k/n$. Consequently,

$$e \cong \sqrt{k/n} \tag{7}$$

The next concept needed is that of an "open angle." For each point p S not on the convex hull, simple triangles with vertices at p will gradually arise in the course of constructing T, until at the end p will be completely surrounded by them. The situation at some intermediate stage in figure 3, in which all of the nondirected edges of p* currently in T are shown.

The closed angles b_1, b_2, ... are the angles within the simple triangles; the open angles a_1, a_2, ... are those that are not. Let a(p,pq) denote the open angle at p through which pq passes.

Lemma 11.1. If Nextut(pq) exists, then $a(p,pq) \geq 60^\circ$ and $a(q,pq) \geq 60^\circ$.

Proof. Let pq" = Nextut(pq). Clearly pq passes through some open angle a′. If $a(p) < 60^\circ$, it contradicts lemma 3, so this is impossible. □

Lemma 11.2. If $pq \in NN(k)$, and $pq'' = \text{Nextut}(pq) \in NN(k')$, then the average $k' = k + d$, where $d < 6$.

Rough Proof. The points nearest p, but at or beyond $|pq|$ from p, will be scattered isotropically around p. One of those located at an open angle will be q′. By lemma 11.1, this will occur after no more than about 6 closer points have been rejected because of end-triangulation. □

By lemma 11.2 and (7), the length of pq" will be no greater than about $((k + d)/n)^{1/2}$ In order for pq" to be crossed by some edge f of the same or smaller length, both vertices of f must be within the "racetrack" area depicted in figure 4.

The area of figure 4 is $(2 + \pi)(k + d)/n$. The expected number of points inside B is $(1 + 2/\pi)(k + d)$. Any triangulation is locally approximately homogenous, so about $3(1 + 2/\pi)(k + d)$ edges in the GT can be expected to be inside B. In a conformal algorithm, these will be mostly edges in $NN(1) \cup NN(2) \cup \ldots NN(k + d)$.

At the time pq" was placed on the ML, it was known that the existing edges -- primarily in NN(1) through NN(k) -- almost certainly did not cross pq". Let us err on the pessimistic side, and assume that (a) almost all the edges in GT have been placed in T at the time pq" reaches the top, and (b) that the populations of NN(1), NN(2), ... NN(k + d) are roughly equally represented in GT, although we know on general grounds that these populations must decrease with increasing index. Even with these assumptions, the expected number of edges inserted inside B in classes NN(k + 1) through NN(k + d) are only d/(k + d) times the total number of edges inside B, or a total of $3(1 + 2/\pi)d$. This is a constant independent of n and k; on the average, then, there is some constant probability that none of the edges inside B will cross pq".

We have now proven that if the probability for pq" to be uncrossed at the time it is placed in the ML is > some nonzero constant, it is > some other nonzero constant when it reaches the top. We now need

Lemma 11.3. If SEN´(pq´,pq) returns FALSE, where pq´ = Nextut(pq), then the probability that pq´ is crossed is less than some constant b > 0.

Proof. The structure of lemmas 9 and 9.1 makes it clear that this must be true provided $r = (|pq'| / |pq|)$ is bounded above by a constant. By lemma 11.2, r is at most the mean ratio of the d + k^{th} nearest-neighbor distance and the k^{th} nearest neighbor distance for some point p. This quantity, by (7), is maximum when k = 1. Therefore r is upper-bounded by the mean ratio of the 7^{th} nearest neighbor distance and the first nearest neighbor distance. This ratio is extremely hard to compute, but it is clearly intensive since it is a ratio, hence constant and independent of n. □

This completes the proof of lemma 11. □

Theorem 1. Algorithm 5 requires space O(n) and average time $O(n^2)$.

Proof. Space is obvious. Lemma 10 leads at once to an O(n) scheme for computing Nextut(pq). Since by lemma 11, the number of d-edges rising to the top of ML is proportional to the number of edges in the triangulation, it follows that only O(n) edges will be placed on the ML during the course of the algorithm. Each of these will require computation of Nextut, insertion on ML, and checking against all $t \in T$. Since each of these requires O(n) steps, we are done. □

We now show that the GT is not optimal even to within some constant factor. The reader´s attention is called to figure 5.

In this figure, a subset A of $m^{3/2}$ points are located on a slightly concave curved line running from (1, ε) to (2, ε). The first and second, reading left to right, are denoted s_1 and s_2; the last by R. Point P, at (0,0), is not "interiorly visible" to points in A, in the sense that it lies above line s_1s_2. There are a subset B of m points which are all on a slightly concave line extending almost vertically from T to Q; all are "interiorly visible" from P. The vertical position of the points b_i in B, indexed top down, is given by

$$y(b_i) = f\sqrt{i} \qquad (8)$$

where f is a constant and f < 1.

The GA will at some time connect b_1 to P. (8) assures that b_2P will later be placed in the plane (rather than b_1s_1), b_3P will be placed in the plane (rather than b_2s_1), etc. Finally, a connection of all points $a \in A$ to Q will be forced.

The result will be that the GT will contain m lines Pb_i, with total length $O(m^{3/2})$, plus the lines connecting the $a \in A$ to Q, with total length $O(m^2)$, plus the lines "in back of" B (because B is concave) with total length $O(\sqrt{m} \log m)$, plus the lines in back of A, with total length O(log m), plus the convex hull, with total length $O(\sqrt{m})$.

There also exists a triangulation which connects T to all $a \in A$, and R to all $b \in B$. The first two categories of the five listed above are replaced by $O(m^{3/2})$ for the former, and $O(m^{3/2})$ for the latter. Hence the ratio of cumulative edge lengths of the GT to a better triangulation is $O(\sqrt{m})$; since $n = O(m^{3/2})$, it follows that the GT can be worse than the optimum triangulation by a factor of at least $O(n^{1/3})$.

REFERENCES

1. Gilbert, P. D. "New Results in Planar Triangulations." MS Thesis, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois.

2. Lloyd, Errol Lynn. "On Triangulations of a Set of Points in the Plane." 18th Annual IEEE Conference Proc. on the Foundation of

Computer Science, Providence, R. I., 1977.

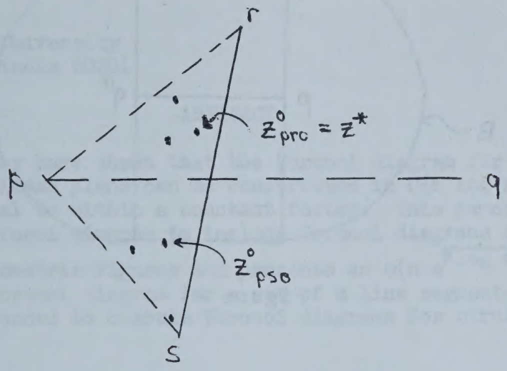

r
$Z^{o}_{prc} = z^{*}$
p
q
Z^{o}_{pso}
s

Figure 1

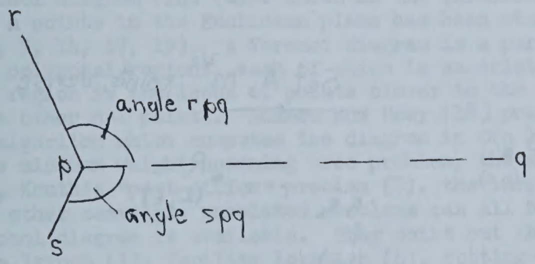

r
angle rpq
p
q
angle spq
s

Figure 2. Showing situation in which step 3 returns FALSE

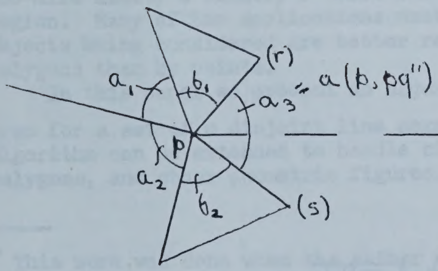

(r)
a_1
b_1
$a_3 = a(p, pq'')$
p
q''
a_2
b_2
(s)

Figure 3

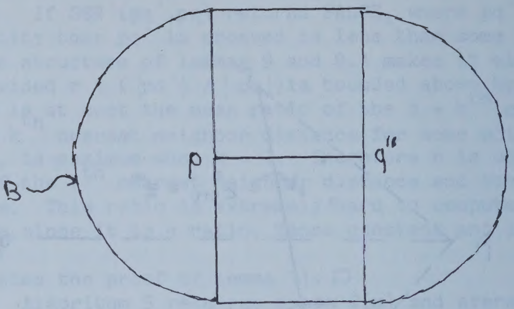


Figure 4

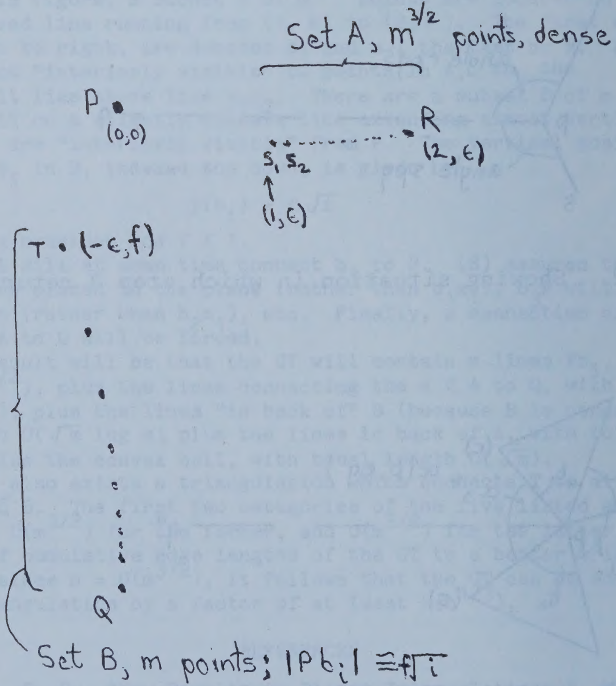

Set A, $m^{3/2}$ points, dense
P
(0,0)
R
(2, ε)
s_1 s_2
(1, ε)
T (-ε, f)
Q
Set B, m points; $|Pb_i| \equiv f\sqrt{i}$

Figure 5

GENERALIZED VORONOI DIAGRAM IN THE PLANE

R. L. DRYSDALE, III*
Dartmouth College
Hanover, New Hampshire 03755

D. T. LEE**
Northwestern University
Evanston, Illinois 60201

ABSTRACT

Shamos and Hoey have shown that the Voronoi diagram for a set of n points in the Euclidean plane can be constructed in O(n log n) time, and that this is optimal to within a constant factor. This paper generalizes the notion of a Voronoi diagram to include Voronoi diagrams for line segments and other geometric figures and presents an $O(n\, c^{\sqrt{\log n}})$ algorithm to construct the Voronoi diagram for a set of n line segments. This algorithm can be extended to compute Voronoi diagrams for circles, polygons, and other figures.

1. INTRODUCTION

The Voronoi diagram [14] (also known as the Thiessen diagram [6, 19]) for a set of n points in the Euclidean plane has been studied by a number of people [8, 9, 14, 17, 19]. A Voronoi diagram is a partitioning of the plane into n polygonal regions, each of which is associated with a given point. Each region is the locus of points closer to the given point than to any of the other n-1 points. Shamos and Hoey [18] present a divide-and-conquer algorithm which computes the diagram in O(n log n) time and show that the minimum weight spanning tree problem, the all-nearest-neighbors problem, Knuth's "post-office" problem [7], the largest empty circle problem, and other seemingly unrelated problems can all be solved quickly once the Voronoi diagram is available. They point out that such problems arise in wire layout [1], facility location [4], cutting-stock and geometric optimization problems [16], clustering problems [5], and contouring problems [2]. A natural question is, "Can the Voronoi diagram be generalized to figures other than points?" Shamos [17] poses the problem of quickly finding the minimum weight spanning tree for a set of circles or line segments. A fast algorithm for computing the Voronoi diagram for a set of circles or line segments would solve this problem. Ignoring the 200-mile limit, a country's territorial waters are precisely its Voronoi region. Many of the applications mentioned above have analogs where the objects being considered are better represented by segments, circles, or polygons than by points.

In this paper we present an algorithm which computes the Voronoi diagram for a set of n disjoint line segments in $O(n\, c^{\sqrt{\log n}})$ time. The algorithm can be extended to handle circles, intersecting line segments, polygons, and other geometric figures.

---

[*] This work was done when the author was at Stanford University, and was supported by the National Science Foundation grant MCS77-05313.

[**] This work was done when the author was with the University of Illinois at Urbana-Champaign, and was supported in part by the National Science Foundation grant MCS76-17321 and in part by the Joint Services Electronics Program under Contract DAAB-07-72-0259.

We start with some definitions. The distance $d(p, s)$ between a point p and a segment s is defined to be the minimum distance between p and any point q in s. The point of s which is closest to p is called the image of p on s. Similarly, the distance $d(s_i, s_j)$ between two segments s_i and s_j is the minimum distance between p in s_i and q in s_j, and the distance $d(p, S)$ between a point p and a set of segments S is the minimum $d(p, s)$ for s in S. The bisector $B(s_i, s_j)$ of two segments s_i and s_j is defined to be the locus of points equidistant from s_i and s_j. The bisector $B(S, T)$ of two sets of segments S and T is the locus of points equidistant from S and T. A bisector $B(s_i, s_j)$ is oriented by imposing a direction on it so that s_i lies to its left and s_j lies to its right. The bisector $B(S, T)$ is oriented in a similar manner.

Given a set S of n line segments s_i, the Voronoi region $V(s_i)$ associated with s_i is defined to be the set of points p in the plane closer to s_i than to any other s_j, $j \neq i$. The Voronoi diagram $VD(S)$ is the set of Voronoi regions $V(s_i)$, for s_i in S. The boundary between two Voronoi regions is called an edge of the Voronoi diagram. Verticies where three or more edges intersect are called Voronoi points. Note that an edge between $V(s_i)$ and $V(s_j)$ is a subset of $B(s_i, s_j)$. Also note that $B(s_i, s_j)$ consists of pieces of lines and pieces of parabolas. Where the images of bisector points on s_i are all the same endpoint of s_i and the images of those same bisector points on s_j are interior points of s_j (or reverse s_i and s_j), the bisector is a piece of a parabola. Where the images are endpoints of both segments or interior points of both segments, the bisector is a piece of a line.

Voronoi regions for points are convex polygons. Voronoi regions for line segments are not even convex. However, they have a similar property. We say that a region R in the plane is generalized-star-shaped with nucleus C if whenever p is in R and q is a closest point to p in C, then $\overline{pq}$ is entirely contained in R.

Lemma 1: Voronoi region $V(s_i)$ is generalized-star-shaped with center s_i.

Lemmas will be stated without proof because of space limitations. The proofs of these lemmas appear in [3] and [11]. One consequence of Lemma 1 is the fact that Voronoi regions for segments are connected.

We are often interested not in the Voronoi diagram itself, but rather in its dual. The dual of a Voronoi diagram is a multi-graph with a node n_i associated with each region $V(s_i)$ in the Voronoi diagram $VD(S)$. There is an arc between n_i and n_j in the dual for each edge between $V(s_i)$ and $V(s_j)$ in $VD(S)$. Figure 1 shows a Voronoi diagram and its dual. Our interest in the dual is a consequence of the following lemmas:

Lemma 2: If S is a set of disjoint line segments and s_j is a nearest neighbor of s_i (i.e., $d(s_i, s_j) = \min(d(s_i, s_k)$ for $k \neq i)$, then n_j and n_i are adjacent in the dual of $VD(S)$.

Lemma 3: Any minimum spanning tree for a set S of disjoint line segments is embedded in the dual of $VD(S)$.

Lemma 4: The dual of a Voronoi diagram on a set S of n disjoint line segments is planar and has at most 3n-5 edges and 2n-3 faces.

These lemmas are interesting because the direct way to find all nearest neighbors or to compute a minimum spanning tree involves computing $n(n-1)/2$ distances. But given the Voronoi dual, only 3n-5 distances need to be computed. This reduction from $O(n^2)$ to $O(n)$ distances means that an algorithm which computes the Voronoi diagram in time faster that $O(n^2)$ will lead to a faster algorithm for solving these problems.

2. CONSTRUCTING THE VORONOI DIAGRAM FOR A SET OF POINTS

Our algorithm is a generalization of the divide-and-conquer scheme first presented by Shamos and Hoey [18] for a set S of n points p_i. We therefore start by describing a version of this algorithm which includes a modification made by Lee [9]. Sort the n points lexicographically on their (x, y) coordinates. Let $L = \{p_1, p_2, \ldots, p_{\lfloor n/2 \rfloor}\}$ and let $R = \{p_{\lfloor n/2 \rfloor + 1}, \ldots, p_n\}$, where $\lfloor a \rfloor$ is the greatest integer in a. Then compute VD(L) and VD(R) recursively. This completes the "divide" part. If we can merge VD(L) and VD(R) in linear time, we can compute VD(S) in O(n log n) time.

The key to the merge step is the curve B(L, R). (See Figure 2.) Everything to the left of B(L, R) is closer to some point in L than to any point in R. Similarly everthing to the right of B(L, R) is closer to some point in R than to any point in L. Therefore if we chop off the part of any region in VD(L) extending to the right of B(L, R) and the part of any region in VD(R) extending to the left of B(L, R), we end up with the regions of VD(S).

We want to construct B(L, R) in linear time. It is a polygonal line with two infinite rays. We begin our construction by finding those rays. The following lemma gives us a way to find them.

Lemma 5: Segments are associated with infinite regions in VD(S) if and only if they lie on the convex hull of S (abbreviated CH(S)).

This lemma suggests computing CH(S) and scanning it to find the two segments connecting points in L with points in R. We can do this by constructing CH(L) and CH(R) recursively when constructing VD(L) and VD(R) and then merging them in linear time using an algorithm invented by Preparata and Hong [12].

Once we have the infinite rays we must connect them. We do this by thinking of an imaginary point starting at infinity and moving up $B(p_4, p_{10})$. The point follows $B(p_4, p_{10})$ until it crosses the border of $V(p_4)$ and finds itself in $V(p_8)$. It is then closer to p_8 than to p_4, so it follows $B(p_8, p_{10})$ until it leaves $V(p_{10})$. In this way it traces out B(L, R), always following the bisector $B(p_l, p_r)$ between a point p_l in L and a point p_r in R. When the point leaves $V(p_l)$, p_l is changed to the point associated with the new region entered. Point p_r is updated similarly.

Because each $V(p_i)$ can have O(n) edges in its boundary in the worst case and we might examine the edges of a single region O(n) times to see where a bisector first intersects the boundary of that region, the naive approach of checking each edge of $V(p_l)$ and $V(p_r)$ to find the first intersection with $B(p_l, p_r)$ could result in $O(n^2)$ work. We avoid this by adopting the following scanning order for edges: Beginning with the point where $B(p_l, p_r)$ enters a region, scan the edges of a region in VD(L) in a counter-clockwise (CCW) direction and the edges of a region in VD(R) in a clockwise (CW) direction. If an edge does not intersect $B(p_l, p_r)$, then discard it. The following lemma can be used to show that no backtracking is needed when using this scanning order.

Lemma 6: Let p_l be in L and p_r be in R and let $B(p_l, p_r)$ be constructed during the merge process. No point of $V(p_l)$ [$V(p_r)$] in VD(L) [VD(R)] which lies to the right [left] of $B(p_l, p_r)$ will be included in the region $V(p_l)$ [$V(p_r)$] in the final diagram. Furthermore, scanning $V(p_l)$ in a CCW direction and $V(p_r)$ in a CW direction will find the first intersection between $B(p_l, p_r)$ and either $V(p_l)$ or $V(p_r)$.

We now count the number of times that we test to see if $B(p_l, p_r)$ intersects an edge. After each test either an edge of VD(L) or an edge of VD(R) is discarded or else a new Voronoi point is created in VD(S). Therefore the total number of tests made in the entire scan is proportional to the total number of edges in VD(L) and VD(R) plus the number of Voronoi points in VD(S). By Lemma 4 this is $O(n)$.

We have not considered the case where B(L, R) intersects a Voronoi point rather than an edge. An added bonus from the CW-CCW scan described above is the fact that it automatically chooses the correct "next" region in this case as well.

3. CONSTRUCTING THE VORONOI DIAGRAM FOR A SET OF SEGMENTS

3.1 From Points to Segments

What happens when we apply this algorithm to a set of segments? The first difficulty arises when sorting the segments. Lexicographical order is not defined for segments. We arbitrarilly decide to sort on the leftmost endpoints.

The CW-CCW scan procedure works for disjoint segments if we modify it slightly. Instead of discarding edges when no intersection is found, we replace them with a single "dummy" edge. For details see [3]. The $O(n)$ time bound on the work required to trace B(L, R) carries over to this case, too. All we have to do is to find the place to start the process.

Figure 3 shows that this is not as simple as it was when we were constructing the Voronoi diagram for points. B(L, R) has broken into four pieces! We have to start the CW-CCW scan four times instead of once. We can find the starting points for the two infinite pieces by constructing CH(L) and CH(R) and merging them just as we did in the points case. The merge process is more complex, but not conceptually difficult (see [3]). However, two of the pieces of B(L, R) are closed curves. We will call these closed curves islands. The convex hull gives us no information about islands.

The authors tried a number of approaches, but found no method for finding all islands in $O(n)$ time. They also tried to compute an order for the segments which would prevent the islands from forming. (The decision to sort on leftmost endpoints was arbitrary.) Finding this order seems to be as difficult as finding the Voronoi diagram itself.

Islands form because segments in L start to the left of but extend to the right of segments in R. These segments in R get trapped in CH(L) and become islands. If we were to cut off and throw away the part of any segment in L that extends to the right of a segment in R we would have no islands. We would also have a much different Voronoi diagram. If instead of throwing away the parts of segments in L that extend to the right of segments in R we made them new members of R we again would have no islands. But this time the Voronoi diagram would be very close to the one we want. Some regions in VD(S) would be broken into two regions in the new diagram, but they could be patched together again quite easily.

The problem with this approach is the fact that the right sub-problem may end up with as many segments as the original problem. The recursion blows up, and in the worst case $O(n^2)$ cuts are made. The usual way to avoid this problem is to split the original problem into a large number of sub-problems. That approach does not help with this particular cutting scheme, but is it really necessary to cut every segment in L which extends to the right of a segment in R? The answer is, "no."

Lemma 7: Cut the segment located directly above the rightmost endpoint r of each segment in S at the intersection point between the segment to be cut and the vertical line through r. (If the segment does not exist or does not contain points on both sides of the vertical line through r then that r does not cause a cut to be made.) Similarly cut the segments below the rightmost endpoint of each segment in S. Call this new set of n' segments S'. Sort the segments s'_i in S' on their leftmost endpoints using the following order: First compare x-coordinates. In case of a tie, a vertical segment is "less than" a segment which is not vertical. In case of a second tie, compare y-coordinates. If $L = \{s'_j, \ldots, s'_k\}$ and if $R = \{s'_{k+1}, \ldots, s'_{n'}\}$ then B(L, R) has no closed curves.

Sketch of Proof: First show that a ray starting at a leftmost point of any segment in L and pointing horizontally to the left will not intersect B(L, R). Then show that a ray starting at a rightmost point of any segment in R and pointing horizontally to the right will either not intersect B(L, R) or else will intersect a higher-numbered segment in R before crossing B(L, R). This means that we can start at any segment and trace a path to infinity without crossing B(L, R). Therefore B(L, R) cannot have any closed curves.

Figure 4 shows what happens to the segments shown in Figure 3 after doing the cutting and sorting described in Lemma 7. It also shows the merge curve B(L, R) that results after dividing the segments into two equal sized sets L and R. Note that the islands have disappeared.

3.2 The Algorithm for Disjoint Segments

Lemma 7 gives us a way to divide the n' segments in S' into f(n) subproblems and then to merge the f(n) Voronoi diagrams into a single diagram for V(S'). For Lemma 7 to hold, R must contain all segments in S' that have numbers higher than the numbers of the segments in L. This leads to a right-to-left merge order. That is, the rightmost two diagrams are merged first, then the third from the right is merged with the combined diagram for the other two, and so on until the leftmost diagram is merged with the combined diagram for the rest of the sub-problems.

We summarize the algorithm:

Step 1: Perform the cutting and sorting described in Lemma 7.

Step 2: Divide the segments in S' into f(n) sets $T_1, T_2, \ldots, T_{f(n)}$, where $T_i = \{s'_{\lfloor (i-1)n/f(n) \rfloor + 1}, \ldots, s'_{\lfloor i \cdot n/f(n) \rfloor}\}$. Recursively construct the Voronoi diagrams $VD(T_i)$. For i running from f(n)-1 down to 1, merge

VD(T_i) with VD($T_{i+1} \cup T_{i+2} \cup \dots \cup T_{f(n)}$). If we call T_i set L and $T_{i+1} \cup T_{i+2} \cup \dots \cup T_{f(n)}$ set R, then we can describe the merge step as follows.

Step 2a: Merge CH(L) and CH(R) to obtain CH(L ∪ R).

Step 2b: Scan CH(L ∪ R) once in a CCW direction. Each time a segment s'_l in L is followed by a segment s'_r in R, use B(s'_l, s'_r) as the starting place for constructing a piece of B(L, R).

Step 2c: Use the CW-CCW scan procedure to construct a piece of B(L, R).

Step 3: Rejoin cut pieces. Find segments with a sub-piece in L and a sub-piece in R and erase the cut. The Voronoi region for the super-piece is the union of the Voronoi regions for the sub-pieces. Similarly update the convex hull.

3.3 Run Time Analysis

The cutting and sorting in Step 1 can be done in O(n log n) time. (The cutting can be implemented by using an AVL tree to keep track of the vertical order of the segments.) The merge in Steps 2a, 2b, and 2c takes O(n) time. These steps are repeated f(n)-1 times, so Step 2 takes O(n·f(n)) time. Step 3 can be done in O(n) time. Letting T(n) be the time required to construct the Voronoi diagram for n segments, we get

$$T(n) = f(n) \cdot T(3n/f(n)) + C_1 \cdot n \cdot f(n) + C_2 \cdot n \cdot \log n.$$

We want to choose f(n) to be the function which minimizes T(n). It turns out that the correct f(n) is

$$f(n) = c^{\sqrt{\log n}}, \text{ for } c > e^{\sqrt{2 \log 3}} \approx 4.40.$$

This function grows slower than n^ϵ for any ϵ, but faster than $(\log n)^p$ for any p. This leads to the solution

$$T(n) = C \cdot n \cdot c^{\sqrt{\log n}} + o(n \cdot c^{\sqrt{\log n}}),$$

where the value of C depends on the value of c.

Theorem: The algorithm described in Section 3.2 computes the Voronoi diagram for a set of n disjoint segments in $O(n\, c^{\sqrt{\log n}})$ time and O(n) space.

4. CONCLUSION

We have presented an algorithm which constructs the Voronoi diagram for a set of n disjoint segments in the plane in $O(n\, c^{\sqrt{\log n}})$ time. This algorithm can be extended to handle intersecting line segments, polygons, circles, and other geometric figures. The necessary modifications are described in [3] and [11]. A complete implementation of the algorithm for line segments which may intersect at endpoints appears in [3]. For this implementation the break-even point between solving the all-nearest-neighbors problem directly and solving this problem by first computing the Voronoi diagram was less than 100 segments in cases where little cutting occurred.

The best lower bound for this problem is $\Omega(n \log n)$, so there is room for improvement in either the algorithm or the bound. Shortly before this conference the authors designed an $O(n (\log n)^2)$ algorithm for circles or line segments. (See [11].) This algorithm finds islands by using the Voronoi diagram VD(L) to find segments in L close to segments in R (see [13]). This algorithm will be presented in another paper.

REFERENCES

1. Breuer, M. A. ed., *Design Automation of Digital Systems*, vol. 1, Prentice-Hall (1974).
2. Davis, J. C. and McCullagh, M. J. eds., *Display and Analysis of Spatial Data*, Wiley (1975).
3. Drysdale, R. L., III, "Generalized Voronoi Diagrams and Geometric Searching", Ph.D. Thesis, Dept. of Computer Science, Stanford University, Oct. 1978.
4. Francis, R. L. and White, J. A., *Facility Layout and Location*, Prentice-Hall (1974).
5. Hartigan, J. A., *Clustering Algorithms*, Wiley Series in Probability and Statistics (1975).
6. Horton, R. E., "Rational Study of Rainfall Data Makes Possible Better Estimates of Water Yield ...", Engineering News-Record, 1917, pp. 211-213.
7. Knuth, D. E., *The Art of Computer Programming*, vol. 3, Addison-Wesley (1973).
8. Kopec, R. J., "An Alternative Method for the Construction of Thiessen Polygons", *Professional Geographer*, 15 (1963), pp. 24-26.
9. Lee, D. T., "On finding k-Nearest Neighbors in the Plane", M.S. Thesis, Coordinated Science Lab., Rep. R-728, Univ. of Ill., Urbana, Illinois, May 1976.
10. Lee, D. T. and Wong, C. K., "Voronoi Diagrams in L_1 (L_∞) Metrics with 2-Dimensional Storage Applications", IBM Res. Rep. RC(6848), Yorktown Heights, N.Y., 1977.
11. Lee, D. T., "Proximity and Reachability in the Plane", Ph.D. Thesis, Dept. of Computer Science, Univ. of Ill., Urbana, Illinois, Sept. 1978.
12. Preparata, F. P. and Hong, S. J., "Convex Hulls of Finite Sets of Points in Two and Three Dimensions", *C. ACM* 20, 2 (1977).
13. Preparata, F. P., "A New Approach to Planar Point Location", submitted for publication.
14. Rhynsburger, D., "Analytic Delineation of Thiessin Polygons", *Geographic Analysis*, 5 (1973), pp. 133-144.
15. Roger, C. A., *Packing and Covering*, Cambridge University Press (1964).
16. Saaty, T. L., *Optimization Problems in Integers and Related Extremal Problems*, McGraw-Hill (1970).
17. Shamos, M. I., *Computational Geometry*, Dept. of Computer Science, Yale Univ., 1977. To be published Springer-Verlag.
18. Shamos, M. I. and Hoey, D., "Closest-Point Problems", *Proc. 16th IEEE Symp. on Foundations of Computer Science*, Oct. 1975, pp. 151-162.
19. Thiessen, A. H., "Preciptitation Averages for Large Areas", *Monthly Weather Review*, 39(1911), pp. 1082-1984.

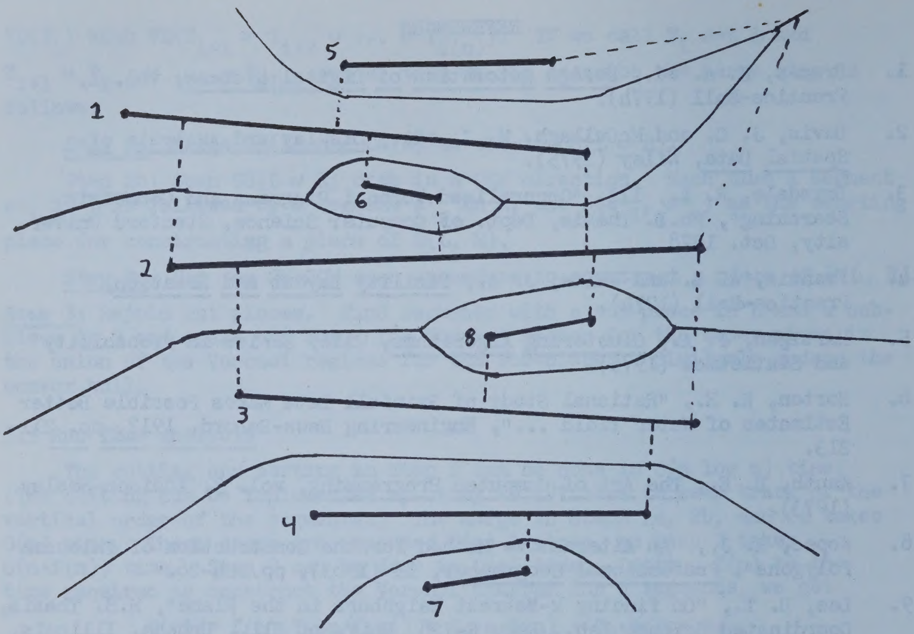


Figure 1. Voronoi Diagram and its Dual

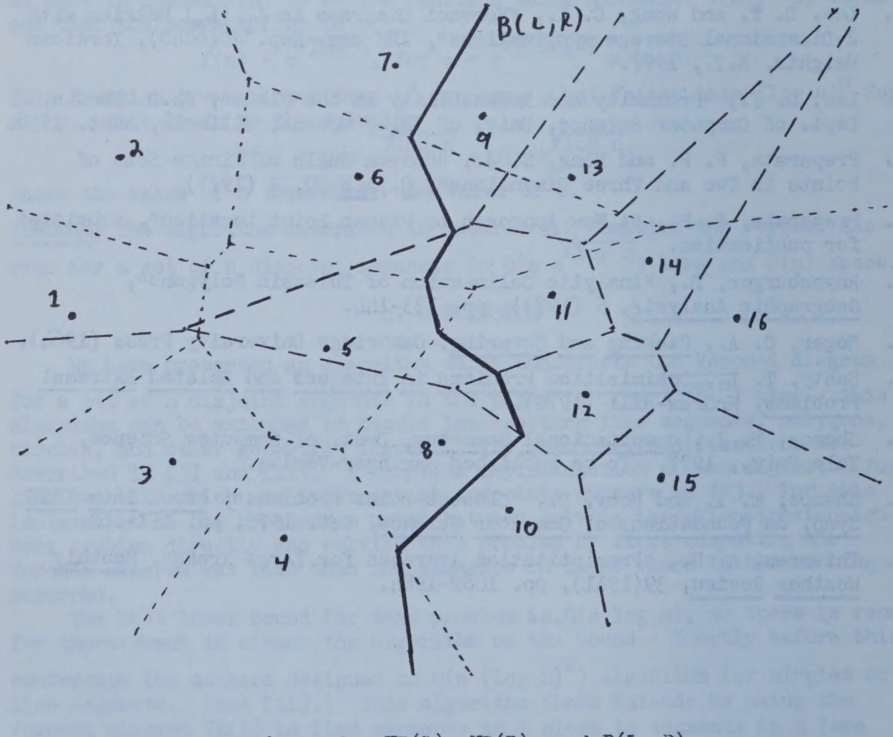

B(L,R)
7.
9
.2
6
13
1.
14
11
16
5
12
3
8
15
10
4

Figure 2. VD(L), VD(R), and B(L, R)

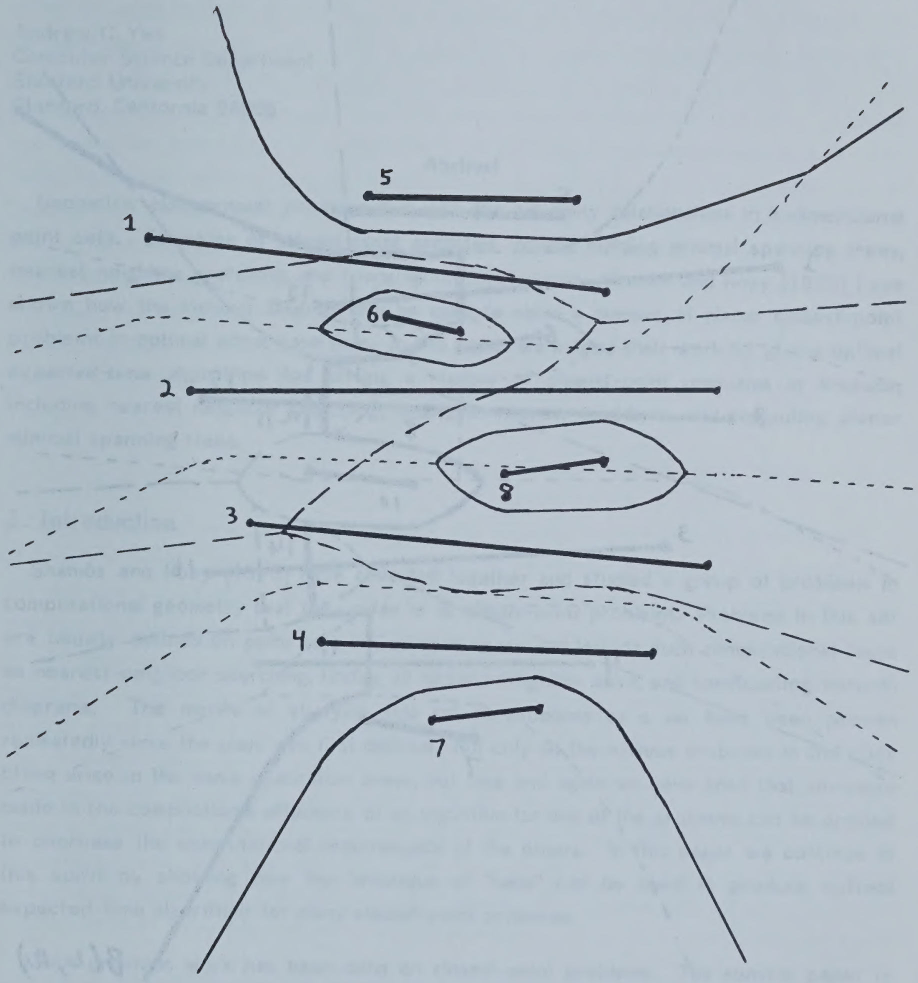


Figure 3. B(L, R) Breaks into Four Pieces

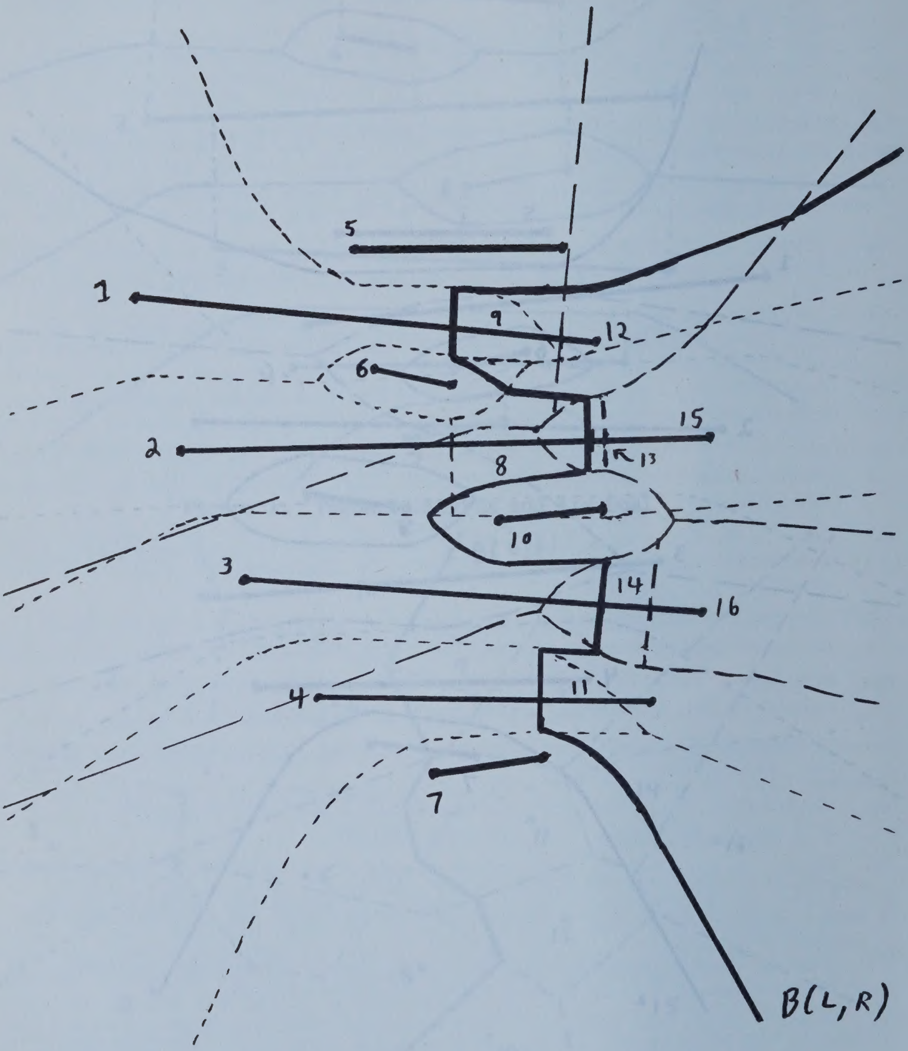

5
1
9
12
6
15
2
13
8
10
3
14
16
4
11
7
B(L,R)

Figure 4. After Cutting B(L, R) has No Islands

OPTIMAL EXPECTED-TIME ALGORITHMS FOR CLOSEST-POINT PROBLEMS

Jon Louis Bentley and Bruce W. Weide
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Andrew C. Yao
Computer Science Department
Stanford University
Stanford, California 94305

Abstract

Geometric *closest-point problems* deal with the proximity relationships in k-dimensional point sets. Examples of closest-point problems include building minimal spanning trees, nearest neighbor searching, and triangulation construction. Shamos and Hoey [1975] have shown how the *Voronoi diagram* can be used to solve a number of planar closest-point problems in optimal worst-case time. In this paper we extend their work by giving optimal *expected-time* algorithms for solving a number of closest-point problems in k-space, including nearest neighbor searching, finding all nearest neighbors, and computing planar minimal spanning trees.

1. Introduction

Shamos and Hoey [1975] have collected together and studied a group of problems in computational geometry that they refer to as *closest-point* problems. Problems in this set are usually defined on point sets in Euclidean space and include such computational tasks as nearest-neighbor searching, finding all nearest-neighbor pairs, and constructing Voronoi diagrams. The merits of studying this set of problems *as a set* have been proven repeatedly since the class was first defined. Not only do the various problems in the class often arise in the same application areas, but time and again we have seen that advances made in the compuational efficiency of an algorithm for one of the problems can be applied to decrease the computational requirements of the others. In this paper we continue in this spirit by showing how the technique of "cells" can be used to produce optimal expected-time algorithms for many closest-point problems.

Much previous work has been done on closest-point problems. The seminal paper in this field is the classic work of Shamos and Hoey [1975] in which the problems are defined and a number of optimal worst-case algorithms for *planar* point sets are given. Algorithms for closest-point problems in higher-dimensional spaces have been given by Bentley [1976], A. Yao [1977], and Yuval [1976]. Randomization algorithms for the closest-pair problem have been given by Rabin [1976] and Weide [1978]; Fortune and Hopcroft [1978] have recently shown that the speedup of the fast closest-pair algorithms was not due to their randomized nature alone but also to the model of computation employed (which allowed floor functions). For additional results on closest-point problems the reader is referred to Preparata and Hong [1977] and Lipton and Tarjan [1977].

In this paper we study closest-point problems from a viewpoint not taken by any of the above authors. We assume that the point sets are randomly drawn from some "smooth" underlying distribution, and then use the cell technique to give fast expected-time algorithms for many closest-point problems. In Section 2 we illustrate this idea by applying it to two fundamental closest-point problems (nearest-neighbor searching and Voronoi diagram construction) under the very restrictive assumption that the points are drawn uniformly from the unit square. In Section 3 we extend these results to other planar closest-point problems and to point sets drawn from a wide class of distributions. In Section 4 we extend our algorithms to problems in Euclidean k-space. Most of the algorithms we present solve a problem on n inputs in expected time proportional to n and all searching structures we give have constant expected retrieval time; these results are therefore optimal by the trivial lower bounds. Conclusions and directions for further research are offered in Section 5.

2. The Basic Algorithms

The problem that is easiest to state and most clearly illustrates the cell method is *nearest-neighbor searching*, sometimes called the *post office problem*. Given n points in Euclidean space, we are asked to preprocess the points in time P(n) in such a way that for a new *query point* we can determine in time Q(n) which point of the original set is closest to it. There are (complicated) structures available for solving the problem with $P(n) = O(n \log n)$ and $Q(n) = O(\log n)$ in the worst case (Lipton and Tarjan [1977]), but one might expect that on the average we can do better. In fact we will see that for a large class of distributions of points, the expected values of P(n) and Q(n) can be made to be O(n) and O(1), respectively. Although we will initially restrict our attention to this apparently simple problem (and the planar case at that), the techniques used also apply to other closest point problems. In particular, we will briefly describe how it is possible to construct the Voronoi diagram (see Shamos and Hoey [1975]) of a planar point set in linear expected time.

We first consider the problem of nearest neighbor searching in the plane, where the points (both the original n points and the query point) are chosen independently from a uniform distribution over the unit square. The idea of the preprocessing step is to assign each point to a small square (*bin* or *cell*) of area C/n, so that the expected number of points in each bin is C. This is easily done by creating an array of size $(n/C)^{1/2}$ by $(n/C)^{1/2}$ that holds pointers to the lists of points in each bin.

When a query point comes in, we search the bin it would be assigned to. If it is empty, then we start searching the cells surrounding it in a spiral-like pattern until a point is found. Once we have one point we are guaranteed that there is no need to search any bin which does not intersect the circle of radius equal to the distance to the first point found and centered at the query point. Figure 1 shows how the spiral search might proceed for the query point in the middle of the circle. Once the point in the northeast neighbor is found, only bins intersecting the circle must be searched. These are marked with x in Figure 1. In order to make this test easy, we suppose that all bins that lie within that distance of the query point in either coordinate are examined, making the number of cell accesses slightly larger than necessary but simplifying the specification of how the appropriate bins are to be found.

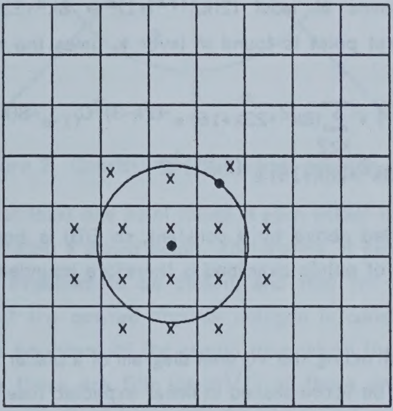


Figure 1. Spiral nearest neighbor search using cells.

It is clear that preprocessing, which consists of assigning each point to the appropriate bin, can be accomplished in linear time if computation of the floor function in constant time is allowed. *This assumption is necessary to solve most of the closest point problems in o(n log n) time because O(n log n) lower bounds are known for many of them in the "decision tree with linear functions" model of computation.* The following theorem shows that spiral search is a solution to the problem we originally posed if the points are uniformly distributed.

> Theorem 1 - If n points are chosen independently from a uniform distribution over the unit square, then the nearest neighbor of a query point can be found in constant expected time using the spiral search.

Proof - We merely need to determine the expected number of bins that must be examined by the algorithm. The effects of dependence among the numbers of points in each bin will become negligible as n grows, so we make the simplification that the expected number of distance computations made is approximately equal to C times the number of bins searched. The exact expected number of distance computations can be determined in a straightforward manner, but only at the expense of unmanageable notation which would serve no useful purpose here. The conclusion from such an exercise is the same as that stated in the theorem.

Let B be the number of bins searched in answering a query when there are n points. If the first bin searched is not empty (this happens with probability $1-(1-C/n)^n \rightarrow 1-e^{-C}$) then we search at most 25 bins (a worst-case bound that cannot be achieved). If it is empty, then we search another "layer" of up to 8 bins centered at the first one; and so on. A point found in one of these bins can be at most $2^{3/2}E$ from the query point, where E is the length of each side of a cell, so the total number of bins searched is at most $(2\lceil 2^{3/2}\rceil+1)^2$, or 49. In general, if the first point is found in the kth layer of the spiral search, we must examine at most $(2\lceil k2^{1/2}\rceil+1)^2 \rightarrow 8k^2+23k+16$ cells. Summing the probability that the first point is found in layer k, times the number of bins examined if that occurs, gives

$$\begin{aligned} E(B) &\rightarrow 25\,(1-e^{-C}) + \sum_{k\geq 2}(8k^2+23k+16)\, e^{-(2k-3)^2C}(1-e^{-8(k-1)C}) \\ &< 25 + \sum_{k\geq 2}(8k^2+23k+16)\, e^{-(2k-3)^2C} \end{aligned}$$

This last sum is bounded above by a constant, so E(B) is bounded by a constant. The expected total number of points examined is therefore bounded by a constant independent of n. ▯

The problem of constructing the Voronoi diagram of a planar point set is somewhat more delicate, but it can still be accomplished in linear expected time under the same assumption of point sets drawn from a bivariate uniform distribution. The basic idea is to search all cells in a relatively small neighborhood of each point in a spiral-like fashion until at least one point is found in each of the eight octants shown in Figure 2, or we give up having examined O(lg n) cells. The *tentative* Voronoi polygon of the center point is that determined by considering just those eight points. Let d be the distance from the center point to the farthest point of its tentative Voronoi polygon. Then no point farther than 2d from the center point can have any affect on the actual Voronoi polygon of that point, which means that the Voronoi polygon of such a point can be constructed by considering only the few (expected constant) number of points which are in the circle of Figure 2.

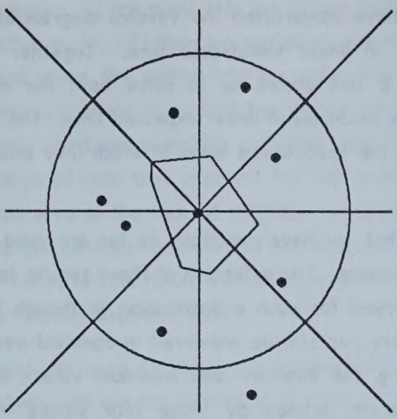


Figure 2. Construction of Voronoi polygon.

In the case that there is at least one point found in each octant before the O(log n) cells are searched, the point is called a *closed* point. Fortunately, it can be shown that all but $O((n \log n)^{1/2})$ points are expected to be closed, and that for each closed point the expected work to construct the desired Voronoi polygon is constant. All other points, most of which are near the boundary of the region from which the points are drawn, are called *open* points. Since there are $O((n \log n)^{1/2})$ of these and each is identified in O(log n) steps, the total work required to identify the open points is $O(n^{1/2} \log^{3/2} n)$.

Once the Voronoi polygons of the closed points are constructed and the open points are idenfified, all of which takes linear expected time, the diagram is completed by applying the O(n log n) Voronoi diagram algorithm of Shamos and Hoey [1975] to the set of open points plus all closed points that share an edge with some open point. The expected size of this set is $O((n \log n)^{1/2})$, so the expected time required by the O(n log n) worst-case algorithm is $O(n^{1/2} \log^{3/2} n)$. A final pass taking at most linear time is then done to remove from this diagram all edges shared by two closed points. The superposition of the remaining diagrams, one from the closed points and the other a modified version of that from the open points and their closed neighbors, is the Voronoi diagram of the entire set.

3. Extensions of the Planar Algorithms

The algorithms of Section 2 can be used to solve a number of planar closest-point problems. Given the fast nearest-neighbor searching algorithm we can easily solve the *all nearest-neighbors* problem (which calls for finding the nearest neighbor of each point) in linear expected time, for point sets drawn from uniform distributions. This is accomplished by preprocessing the n points in linear time and then doing n searches, each of expected constant cost. Once we have found all nearest neighbors we can easily find the *closest pair* in the set by taking the minimum of the n distances. Shamos and Hoey [1975] have shown that once we have constructed the Voronoi diagram of a point set we can solve many other problems in linear worst-case time. Together with the Voronoi diagram algorithm of Section 2 this allows us to solve both the *minimal spanning tree* and *Delaunay triangulation* problems in linear expected time. The details of these algorithms, together with some of the applications areas in which they arise, are discussed by Shamos [1978].

All of the results that we have described so far are valid only for point sets drawn uniformly on the unit square. The extension of these results to unknown distributions is a bit tricky. If we proceed for such a distribution as though it were uniform over some bounded region, a query can still be answered in constant expected time. The cells are chosen by first finding the minimum and maximum values in both x and y and then partitioning the rectangle defined by those four values into a number of squares proportional to n. The resulting cells can be represented by a two-dimensional array and our previous algorithms can operate as before. The only restriction on the underlying distribution required to achieve constant expected time is that it satisfy a condition similar to but more restrictive than a *Lipschitz condition.*

> Theorem 2 - Let n points be chosen independently from the distribution F(x, y) over a bounded region of the plane, where F satisfies the condition that there exists constants $0<C_1\le C_2$ such that for any region of area A, the probability assigned to A by F lies between C_1A and C_2A. (Alternatively, F has a density with respect to Lebesgue measure that is bounded above and bounded away from zero.) Then the same algorithm that was used in the uniform case answers a query in constant expected time.

Proof - The proof of Theorem 1 is easily modified to show that the expected number of distance computations is at most

$$C_2 \left[25 + \sum_{k\geq 2} (8k^2+23k+16)\, e^{-(2k-3)^2 C_1} \right]$$

which is bounded by a constant independent of n. The only difficulties that arise are caused by the possibly irregular boundary of the region from which the points are chosen. Since the region and distribution are fixed and independent of n, it is straightforward to show that even while the bound above holds for points "sufficiently far from the boundary", the only modification required for points "near the boundary" is that the number of comparisons may be larger by a constant factor. Similar arguments show how the above methods can be applied to give linear expected-time algorithms for all of the closest-point problems mentioned above. ▯

4. Extensions to Higher Dimensions

In the previous section we showed how the algorithms of Section 2 can be used to solve a number of problems with inputs drawn from a wide variety of planar distributions; in this section we will see how the basic results can be extended to solve closest-point problems in k-dimensional space. If the point sets are drawn independently and uniformly from the unit hypercube (that is, $[0, 1]^k$) then we can divide the hypercube into n/C cells, each of side $(n/C)^{1/k}$, and place the points into the cells in a processing phase. To perform a nearest neighbor search we generalize the spiral search of Section 2, starting at the cell holding the query point and searching outwards until we find a non-empty cell. At that point we must search all cells that intersect the ball centered at the query point with radius equal to the distance to the nearest neighbor found so far. Arguments similar to those used in Section 2 can be used to show that the expected work performed in this search is constant. Once we have this result we can solve both the all nearest neighbors and closest pair problems in linear expected time.

The cell method can also be used to solve the k-dimensional minimal spanning tree problem in fast expected time. We use the method of A. Yao [1977], which calls for finding the nearest neighbor of each of the n points in each of some critical number of generalized orthants. Yao has shown that the resulting graph is a supergraph of the minimal spanning tree of the point set. Since that graph contains a number of edges linear in n (for any fixed dimension k), the minimal spanning tree can be found in O(n log log n) time by the use of Yao's [1975] algorithm. Since the "nearest neighbor in orthant" searching can be accomplished in constant expected time for each point, the total expected running time of this algorithm is O(n log log n).

It appears to be a very difficult task to use the cell method to construct k-dimensional Voronoi diagrams in fast expected time. Generalizing the method of Section 2 allows us to find the Voronoi polytopes of all closed points in linear expected time, but at that point there still remains $O((n \log n)^{1-1/k})$ open points. Since no fast algorithms are known for constructing Voronoi diagrams in k-space, it is not clear how to find the true Voronoi diagram. Notice, however, that we have found the Voronoi polytopes of almost all the points (that is, the ratio of open points to n approaches zero as n grows). This technique can be used for "Voronoi polytope searching", which asks for the actual Voronoi polytope containing the query point--our method will succeed in constant time with probability approaching one.

The algorithms we have described above have all been for points drawn uniformly in $[0, 1]^k$ Methods analogous to those used in Section 3 can be used to show that the algorithms can be modified to work for point sets drawn from any distribution over some bounded region with density bounded above and below.

5. Conclusions

In this paper we have seen a number of algorithms for solving multidimensional closest-point problems. The algorithms were all based on the simple idea of cells, and were analyzed under the assumption that the points were drawn from some underlying "smooth" distribution. All of the searching methods we described have linear preprocessing costs and constant expected searching costs; all of the algorithms (with the exception of k-dimensional minimal spanning trees) have linear expected running time. It is clear that these algorithms achieve the trivial lower bounds and are therefore optimal. Although we have described the algorithms primarily as theoretical devices (sacrificing efficiency for ease of analysis), they can be implemented on a random access computer very efficiently. Weide [1978] has described how the empirical cumulative distribution function can be used to decrease the constants of the running times even further. Although the searching structures that we have described in this paper are inherently *static*, they can be modified to become *dynamic*. Specifically, for all of the searching problems we have described we can give dynamic structures in which a sequence of n insertions and deletions in O(n) time. At any point in this sequence it is possible to perform a search in constant expected time, and the storage used by the structure is proportional to the number of elements currently stored.

Much further work remains to be done in developing fast expected-time algorithms for closest-point problems. Can the expected complexity of computing minimal spanning trees in k-space be reduced from O(n lg lg n) to O(n)? A particularly important problem is to extend our results from bounded distributions to unbounded distributions (the multivariate normal, for example). It appears that new algorithms will have to be developed for this problem, taking special care of "outliers". Another very interesting open problem is to describe precisely how much of the efficiency of our algorithms is gained from probabilistic assumptions and how much is gained by use of the floor function. (The recent paper of Fortune and Hopcroft [1978] shows that floor can be used to speed up the computation of closest pair without making the randomization assumptions of Rabin [1976] and Weide [1978].)

Acknowledgements

Helpful conversations with James B. Saxe and Professor Michael Ian Shamos are gratefully acknowledged.

References

Bentley, J. L. [1976]. Divide and conquer algorithms for closest point problems in multidimensional space, Ph.D. Thesis, University of North Carolina, Chapel Hill, North Carolina.

Fortune, S. and J. E. Hopcroft [1978]. A note on Rabin's nearest-neighbor algorithm, Cornell Computer Science Report TR78-340.

Lipton, R. J. and R. E. Tarjan [1977]. "Application of a planar separator theorem," 18th FOCS (Oct. 1977), IEEE, pp. 162-170.

Preparata, F. P. and Hong, S. J. [1977]. "Convex hull, of finite sets of points in two and three dimensions," *CACM 20*, 2 (February 1977), pp. 87-93.

Rabin, M. O. [1976]. "Probabilistic algorithms," in *Algorithms and complexity: New directions and recent results*, pp. 21-39, (Ed. J. F. Traub), Academic Press, 1976.

Shamos, M. I. and D. Hoey [1975]. "Closest-point problems," 16th FOCS (Oct. 1975), IEEE, pp. 151-162.

Shamos, M. I. [1978]. Computational geometry, Ph.D. Thesis, Yale University, New Haven, Connecticut.

Weide, B. W. [1978]. Statistical Methods in Algorithm Design and Analysis, Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, PA. To appear.

Yao, A. C. [1975]. "An O(|E| log log |V|) algorithm for finding minimum spanning trees," *Information Processing Letters 4*, 1 (September 1975), pp. 21-23.

Yao, A. C. [1977]. "Fast algorithms for finding minimum spanning trees in k dimensions," *Proceeding Fifteenth Annual Allerton Conference on Communication, Control, and Computing* (September 1977) pp. 553-556.

Yuval, G. [1976]. "Finding nearest neighbors," *Information Processing Letters* 5, 3 (August 1976), pp. 63-65.

PARALLEL SOLUTION OF NONLINEAR INITIAL AND MULTIPLE POINT BOUNDARY VALUE PROBLEMS

Fernando L. Alvarado
Department of Electrical and Computer Engineering
The University of Wisconsin-Madison
Madison, Wi. 53706

ABSTRACT

The numerical method presented in this paper permits the solution of differential initial and boundary value problems by trapezoidal integration in a time of order $\log_2 T$, where T is the number of discrete steps required for the solution. The number of required parallel processors is T/2. Linear and nonlinear examples are presented. The classical trapezoidal integration algorithm is compared to the new parallel trapezoidal algorithm. Overall conclusions indicate that the parallel algorithm is always much faster and sometimes has better convergence characteristics. Potential limitations of the method are also discussed.

INTRODUCTION

Implicit trapezoidal integration has proven to be a most reliable and elegant method for the numerical solution of practical systems of differential equations. Perhaps its most salient property is its numerical stability, which allows the trouble-free analysis of stiff systems of equations [1]. For this and other reasons it has become the method of choice for the analysis of a variety of important practical problems, such as transient stability [2-4] and the calculation of electromagnetic transients [5,6]. The advent of inexpensive computational capabilities, as well as the need for faster real time solutions for certain problems, has generated a great interest in the creation of techniques and algorithms suitable for parallel processing, as evidenced by a variety of symposia and conferences on the subject in recent years [7-11].

The approach to parallel processing followed in this paper is the identification of truly independent tasks that may be executed simultaneously by independent processors [12]. In this regard, a problem that has received a great deal of attention is the solution of algebraic linear equations. Results so far indicate that a good approach for parallel processing of sparse sets of linear equations is by arranging the matrix into a "Blocked Bordered Form" [13,14,15]. For full matrices, a promising alternative is the use of vector or arrays processors [11].

Nowhere are the potential payoffs from parallel processing greater than for the solution of transient problems described by differential equations, such as the transient stability problem or the numerical calculation of elecromagnetic transients [5,6]. Not only are the times of solution usually very large due to the large number of time steps ordinarily required, but also the process may appear to be inherently sequential, requiring the analysis of the system at one step before proceeding to the next. Admittedly, the parallel solution of linear algebraic equations can enhance solution times by permitting faster solutions within any one time step. However, as this paper shows, even greater gains are possible by looking at the entire global problem and achieving parallelism between the different time steps themselves. Later sections show that the solution by trapezoidal integration for $T = 2^\tau$ time steps can be formulated as a single problem involving $T \cdot n$ equations in $T \cdot n$ unknowns, where n is the number of state variables. These equations can then be solved in τ steps provided T/2 processors are available. This, by the way, is an efficient manner of utilizing T/2 processors, since the "speedup factor" is T/log T, and this value tends to T for large T; provided, of course, that each of the parallel steps requires no more effort than each of the serial steps.

The problem of parallel solutions of differential equations has also been studied by Miranker and Liniger [17] and Worland [18] where the use of multi-step predictor-corrector formulas is analyzed.

LINEAR INITIAL VALUE PROBLEM

Consider the following n-th order differential equation:

$$\dot{x} = Ax + f(t) \qquad x(0) = x_o \qquad (1)$$

Using trapezoidal integration the value of x at t+Δt can be determined in terms of x(t+Δt) and x(t). Let the subscript k denote quantities at time t and k+1 quantities at t+Δt .

$$x_{k+1} = x_k + \frac{\Delta t}{2}[A\, x_k + f_k + A\, x_{k+1} + f_{k+1}] \tag{2}$$

This equation can be easily re-arranged as follows:

$$(I-\frac{\Delta t}{2}A)\, x_{k+1} = (I+\frac{\Delta t}{2}A)\, x_k + \frac{\Delta t}{2}(f_k+f_{k+1}) \tag{3}$$

Sequential solution of (3) for k+1, starting with k = 0, yields the desired approximate solution of (1). If n is large and A sparse this solution is generally obtained by LU factorization of [I - (Δt/2)A] by means of well known sparsity techniques. No iterations are required, and if A is time invariant only a single factorization is required, a most desirable situation from the point of view of overall computational complexity [16].

It is, however, possible to solve for the x vectors at T time steps simultaneously. For reasons to become apparent shortly it is best to choose T such that $T = 2^\tau$, τ integer. A set of T vector equations (3) can be written:

$$\begin{aligned}
(I-\tfrac{\Delta t}{2}A)x_1 &= (I+\tfrac{\Delta t}{2}A)x_o + \tfrac{\Delta t}{2}(f_o+f_1)\\
(I-\tfrac{\Delta t}{2}A)x_2 - (I+\tfrac{\Delta t}{2}A)x_1 &= \tfrac{\Delta t}{2}(f_1+f_2)\\
&\vdots\\
(I-\tfrac{\Delta t}{2}A)x_T - (I+\tfrac{\Delta t}{2}A)x_{T-1} &= \tfrac{\Delta t}{2}(f_{T-1}+f_T)
\end{aligned} \tag{4}$$

Define a new vector $\underline{x}$ as follows:

$$\underline{x}^t = (x_1, x_2, \ldots, x_T)$$

Clearly, $\underline{x}$ can be treated as a vector of T unknowns. Equation (4) can be written in a more compact manner as follows:

$$\underline{A}\ \underline{x} = \underline{b} \tag{5}$$

where:

$$\underline{a}_{ij} = \begin{cases} I-\frac{\Delta t}{2}A & \text{for } i=j \\ -(I+\frac{\Delta t}{2}A) & \text{for } i=j+1 \\ 0 & \text{otherwise} \end{cases} \qquad \underline{b}_i = \begin{cases} (I+\frac{\Delta t}{2}A)x_i + \frac{\Delta t}{2}(f_i+f_{i-1}) & \text{for } i=1 \\ \frac{\Delta t}{2}(f_i+f_{i-1}) & \text{for } i\neq 1 \end{cases}$$

Figure 1 illustrates in a compact manner the topology of $\underline{A}$ for T = 16 (τ = 4) when the equations and variables are left in their natural order (1,2,...,T). T matrix equations are in a lower diagonal form. Each one of the T diagonal blocks may be factored independent of all others, whether the factorization is carried into the ℓ terms or not. This in itself is an important observation, since it is well known that the factorization step is usually the most crucial step during the manipulation of sparse ma-

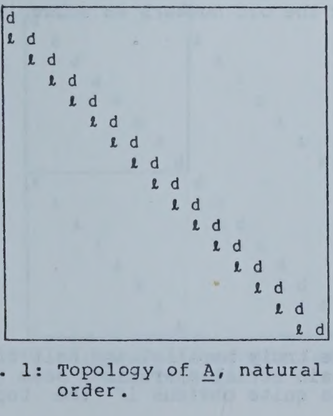


Fig. 1: Topology of $\underline{A}$, natural order.

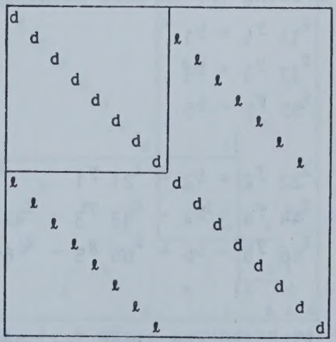


Fig. 2: Topology of $\underline{A}$ after odd-even permutaion.

trices [16,17]. Of course for the linear time invariant case this issue is completely irrelevant, since all diagonal elements d are identical. However, for the time varying and nonlinear cases this advantage alone may justify the use of up to T parallel processors to achieve gains during factorization. Another observation of interest is that it is not necessary to carry out the factorization of the diagonals d into the lower diagonal blocks ℓ [18]. This is possible if the equations in (5) are written as:

$$\begin{aligned} d\,x_1 &= b_1 \\ d\,x_2 &= b_2 - \ell\,x_1 \\ &\;\vdots \\ d\,x_T &= b_T - \ell\,x_{T-1} \end{aligned} \tag{6}$$

which is precisely what is normally done for serial solutions.

Unfortunately, although the factorization step has become highly parallel, the same is not true of the required forward and back substitution steps. As clearly evidenced in (6) or any variation of (6), the calculation of x_{k+1} must await for the value of x_k . The question naturally arises as to whether it is possible to overcome this limitation and perform this step too in parallel. The answer is yes, and it is based on a re-ordering of the equations in (5). However, unlike other re-ordering schemes, the objective is not to reduce fill in (there is none for the natural ordering) but to maximize parallelism during forward substitution. This can be accomplished by a diagonal re-ordering in such a way that $\underline{A}$ becomes a bordered matrix. A most remarkable fact is that this re-ordering will result in fill in where there was previously none. Nevertheless, the overall computational effort will be lower due to the ability to use parallel processors.

In simple terms, the reordering algorithm involves re-numbering all odd rows and columns ahead of all even rows and columns. The results of using this re-ordering on the matrix in Figure 1 are illustrated in Figure 2. It can be observed that it is still possible to factor the first T/2 (8 in this case) blocks independent of each other. However, the factorization must now be carried into the term and will produce a fill in term in the lower right hand corner of $\underline{A}$ as illustrated in Figure 3. Once the factorization is completed, the block-wise substitution step can be performed completely in parallel for the first T/2 blocks by T/2 independent processors, and performed on the remaining T/2 blocks by an additional T/2 processors. This can be more clearly evidenced by re-writing (5) as follows:

$$\mathrlap{\overbrace{\phantom{\underline{L}\,\underline{U}}}^{\underline{A}}}\underline{L}\,\underbrace{\underline{U}\,\underline{x}}_{\underline{y}} = \underline{b} \tag{7}$$

and solving for y from:

$$\underline{L}\;\underline{y} = \underline{b} \tag{8}$$

Writing (8) in more detail, retaining the old numbers as subscripts:

$$\begin{aligned} \ell_{11}\,y_1 &= b_1 \\ \ell_{33}\,y_3 &= b_3 \\ \ell_{55}\,y_5 &= b_5 \\ &\;\vdots \\ \ell_{22}\,y_2 &= b_2 - \ell_{21}\,y_1 \\ \ell_{44}\,y_4 &= b_4 - \ell_{43}\,y_3 - \ell_{42}\,y_2 \\ \ell_{66}\,y_6 &= b_6 - \ell_{65}\,y_5 - \ell_{64}\,y_4 \\ &\;\vdots \end{aligned} \tag{9}$$

Although the first T/2 operations are truly parallel and half of the last T/2 are also clearly parallel, certain serial operations seem to remain. The solution to this dilemma becomes quite obvious if the topology of the lower right hand corner of A after its first reordering is observed (Figure 3). It, too, appears to have a lower banded structure due to the peculiar nature of the fill-in. The re-ordering can then be recursively applied to the last T/2, then T/4 and so on rows and columns. The result of this reordering for T = 16 after fill-in due to factorization is illustrated in Figure 4. This time the detailed general expression for (8) after re-ordering and factorization becomes, retaining the original numbers as subscripts:

$$
\begin{array}{ll}
T/2 & \left\{ \begin{array}{l} \ell_{11}\, y_1 = b_1 \\ \ell_{33}\, y_3 = b_3 \\ \quad\vdots \end{array} \right. \\
T/4 & \left\{ \begin{array}{l} \ell_{22}\, y_2 = b_2 - \ell_{21}\, y_1 \\ \ell_{66}\, y_6 = b_6 - \ell_{65}\, y_5 \\ \quad\vdots \end{array} \right. \\
T/8 & \left\{ \begin{array}{l} \ell_{44}\, y_4 = b_4 - \ell_{43}\, y_3 - \ell_{42}\, y_2 \\ \ell_{12,12}\, y_{12} = b_{12} - \ell_{12,11}\, y_{11} - \ell_{12,6}\, y_6 \\ \quad\vdots \end{array} \right. \\
 & \quad\vdots
\end{array}
\tag{10}
$$

The entire solution process requires τ steps (4 in the example), all of which require at most T/2 processors. The k-th step requires the calculation of $T/2^k$ values of y, and for each value of y there are (k-1) multiplications/ subtractions required. However, partial processing of all of the remaining $T/2^k$ rows is always possible since T/2 processors are available. This circumvents an otherwise undesirable serialization of the calculations. The total number of serial steps is therefore merely τ.

Identical arguments to those presented for the forward substitution step can also be made for the back substitution process. This process also requires τ serial steps, and will not be covered in detail.

The algorithm presented in this section is not the only parallel algorithm of logarithmic order available for the processing of blocked lower banded matrices. Stone [19] has presented an algorithm for the processing of scalar tridiagonal matrices using a technique called recursive doubling. A slight extension of Stone's algorithm would be applicable to the solution of (5). Other researchers that have reached somewhat similar results are Buneman [20], Buzbee et al. [21] and Swarztrauber [22]. These last 3, however, are concerned with a more specialized form of (5), namely, a block triangular form where the "ℓ" matrices are diagonal and d is usually tridiagonal. It is this author's belief that the the algorithm presented in this paper has conceptual advantages and a wider field of applicability than these former efforts.

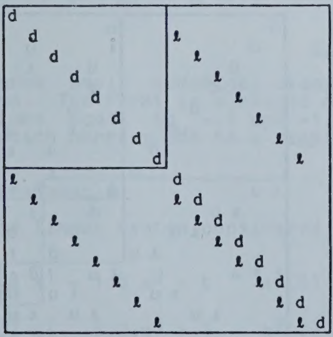


Fig. 3: A after permutation and fill-in.

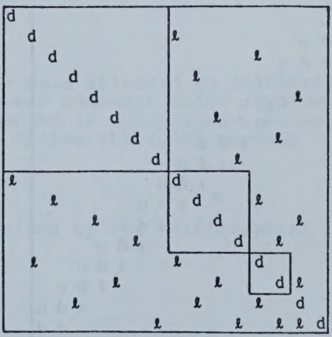


Fig. 4: A after recursive permutation and fill-in.

INITIAL VALUE NONLINEAR PROBLEM

Consider next a set of nth order nonlinear differential equations:

$$\dot{x} = f(x,t) \qquad x(0) = x_0 \tag{11}$$

Application of the trapezoidal rule between t and t+Δt yields:

$$x_{k+1} = x_k + \frac{\Delta t}{2}[f(x_k,t) + f(x_{k+1},t+\Delta t)] \tag{12}$$

In general, (12) will be an implicit equation on x_{k+1}. It can be re-arranged into the following format:

$$g_{k+1}(x_{k+1}) = b_k(x_k) \tag{13}$$

If x_k is known, equation (13) represents a nonlinear algebraic equation on x_{k+1}. The usual method for solving equations of this type is Newton's method [4,24,25]. For notational convenience let x denote x_{k+1} and drop all other subscripts from (13). Then (13) can be expressed as:

$$\varepsilon = g(x) - b = 0 \tag{14}$$

An initial estimate x^0 for x can be improved upon from:

$$x^{i+1} = x^i - \Delta x \qquad J^i \, \Delta x = \varepsilon(x^i) \qquad J^i = \left.\frac{\partial g}{\partial x}\right|_{x^i} \tag{15}$$

The coefficient J^i is the well known Jacobian matrix. For large systems it is frequently sparse. The solution of (11) can proceed sequentially over any number T of discrete time steps. At each time step a Newton solution of the nonlinear algebraic equation (14) is required to determine x_{k+1}. This step usually requires 3 or 4 iterations.

As in the linear case, it is possible to formulate the solution of the equations (13) over several time steps $T = 2^\tau$, τ integer. A set of equations like those in (13) can be written:

$$\begin{aligned} &g_1(x_1) = b_1(x_0) \\ &g_2(x_1,x_2) = b_2 \\ &\quad\vdots \\ &g_T(x_{T-1},x_T) = b_T \end{aligned} \tag{16}$$

These equations can be written in a more compact manner simply as:

$$\underline{\varepsilon} = \underline{g}(\underline{x}) - \underline{b} = 0 \tag{17}$$

This shorthand notation merely represents T·n nonlinear equations in T·n unknowns. Its solution yields all the values of the state variables x for all T time steps. In principle this equation can be also solved by Newton's method, as before. An estimate for $\underline{x}$ can be successively improved upon by solving the equation:

$$\underline{J}\Delta\,\underline{x} = \underline{\varepsilon} \qquad \text{where} \quad \underline{J} = \partial\underline{g}/\partial\underline{x} \tag{18}$$

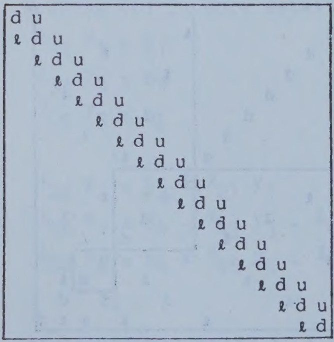


Fig. 5: Topology of $\underline{A}$ for TPBVP.

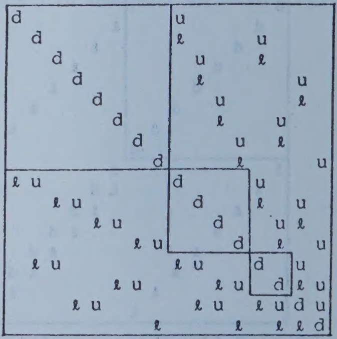


Fig. 6: $\underline{A}$ after recursive permutation and fill-in.

Although this Jacobian is of large dimension, it has a block-wise topological structure which is identical to the one of $\underline{A}$, as depicted in Figure 1. That is, it is a lower banded matrix. Furthermore, since the number of iterations required for the solution of nonlinear equations by Newton's method tends to be independent of system size, it can be expected that the solution of (17) can in fact also be accomplished in a small number (3 or 4) of global iterations. This expectation has been demonstrated during experiments performed by the author. Perhaps the only readily apparent objection is that although good initial estimates for x_1 are available, the estimates for x_k for large k are much less certain. This can conceivably increase the number of required iterations and maybe even lead to non-convergence. Limited experimentation has shown that this is not a major problem. Furthermore, later on an efficient technique for obtaining "good" initial estimates for all k is outlined.

As in the linear case, it is possible to perform the forward and back substitution steps in parallel by T/2 processors if a suitable block-wise re-ordering of J is performed. This re-ordering process was discussed in detail in the previous section. If the number of iterations is assumed to be roughly the same, the "speedup factor" of the complete process is therefore T/τ .

BOUNDARY VALUE PROBLEMS

The technique described in the last two sections is not restricted to initial value problems. In fact, quite arbitrary nonlinear multi-point boundary value problems can be solved. The extension to the more general boundary value problem will be illustrated by considering a simple linear Two Point Boundary Value Problem. Extensions to more general cases should be readily apparent. Let

$$\dot{x} = A\,x + f(t) \qquad x^a(0)=x_o^a \quad x^b(T) = x_T^b \tag{19}$$

Application of the trapezoidal rule, as in equations (3) and (4), yields a set of equations:

$$\begin{aligned}
(I-\tfrac{\Delta t}{2}A)x_1 - (I_b+\tfrac{\Delta t}{2}A_b)x_o^b &= (I_a+\tfrac{\Delta t}{2}A_a)x_o^a + \tfrac{\Delta t}{2}(f_o+f_1)\\
(I-\tfrac{\Delta t}{2}A)x_2 - (I+\tfrac{\Delta t}{2}A)x_1 &= \tfrac{\Delta t}{2}(f_1+f_2)\\
&\vdots\\
(I_a-\tfrac{\Delta t}{2}A_a)x_T^a - (I+\tfrac{\Delta t}{2}A)x_{T-1} &= (I_b+\tfrac{\Delta t}{2}A_b)x_T^b + \tfrac{\Delta t}{2}(f_{T-1}+f_T)
\end{aligned} \tag{20}$$

Where I_a and A_a denote the first a columns of I and A and I_b and A_b denote the last b columns. Re-writing in a more compact manner:

$$\underline{A}\ \underline{x} = \underline{b} \qquad \text{where } \underline{x} = (x_o^b, x_1, x_2, \ldots, x_T^a) \tag{21}$$

The topology of A is illustrated in Figure 5. Notice that both u and ℓ contain, of necessity, zero columns. The matrix $\underline{A}$ has, therefore, acquired a block-tridiagonal topology. Fortunately, the same re-ordering that permitted the parallel solution of the lower trangular matrix also permits the parallel solution of a tridiagonal matrix. Figure 6 illustrates the topology of $\underline{A}$ after factorization and fill-in.

EXAMPLES

Three small numerical examples have been selected to illustrate the algorithm. The first is a second order linear somewhat stiff system with eigenvalues equal to -.1 and -1. The second is a nonlinear second order system which corresponds to a simple power system stability problem.

a) Linear Example

The linear system considered is described by the differential equations:

$$\begin{aligned}
\dot{x}^1 &= -2.8\,x^1 - 1.8\,x^2 + t \qquad x^1(0) = 0\\
\dot{x}_2 &= -7.2\,x^1 - 8.2\,x^2 + 1 \qquad x^2(0) = 0
\end{aligned} \tag{22}$$

Consider the solution of (22) by trapezoidal integration with $\Delta t=.05$ and T=8. The resultant differece equations can be solved serially in 8

Table I: Serial vs. Parallel Trapezoidal Integration, Linear Example.

t		t_{max}	Steps Required Serial	 Parallel	Speedup
.05	4	.8	16	4	4
	5	1.6	32	5	6.4
	6	3.2	64	6	10.7
.1	3	.8	8	3	2.67
	4	1.6	16	4	4
	5	3.2	32	5	6.4

steps. Alternately, a set of 16 linear equations in 16 unknowns can be formulated. This set of 16 equations can then be solved in parallel by performing and odd-even recursive permutation. The results in both cases are identical. Table I summarizes a comparison of both solution techniques for these and other values of t and T.

b) Nonlinear Example

The nonlinear example considered corresponds to the following differential equation:

$$\dot{\delta} = \omega \qquad (0) = 0$$
$$\dot{\omega} = -\omega - 2\sin\delta + 1 \qquad (0) = 0 \tag{23}$$

Discretization of (23) and application of implicit integration yields:

$$\varepsilon_k^{\delta} = \delta_{k+1} - \frac{\Delta t}{2}\omega_{k+1} - \delta_k - \frac{\Delta t}{2}\omega_k = 0$$
$$\varepsilon_k^{\omega} = (1+\frac{\Delta t}{2})\omega_{k+1} + \Delta t \sin\delta_{k+1} - (1-\frac{\Delta t}{2})\omega_k + \Delta t \sin\delta_k - \Delta t \tag{24}$$

As mentioned earlier, the parallel solution algorithm involves the construction, re-ordering and factorization of a larger Jacobian matrix $\underline{J}$ whose elements are either zero blocks or full 2x2 blocks. Parallel solutions also require the initialization of x_1 and x_2 over all T time steps. Two approaches to the initialization of $\underline{x}$ were considered. The first approach involved initialization of all values of x_k with values equal to x_0. This initialization can be called a "flat start", and it is quite simple to implement. Limited experience with it shows that it results in excellent convergence characteristics.

It is conceivable, however, that certain problems might have convergence difficulties under certain circumstances. For this reason a second more complex way of initializing $\underline{x}$ was developed. Prior to initialization of the problem, the same original problem is solved over the interval in question but with a larger Δt. For example, an initial interval equal to to 2 Δt can be This solution can either be performed serially or, better yet, it can also be performed in parallel. The results of this solution with a large Δt can then be interpolated and used as initial values on the actual problem. The interpolation process can easily be done in parallel. This initialization approach will be referred to as "sequential start". Of course, if a parallel approach is used on the initialization run, that run itself must be initialized.

A particularly promising choice of initialization interval is $2^{\tau/2}$. Thus, it is possible to conceive a recursive initialization algorithm which would require the solution of a sequence of problems with 2^1, 2^2, 2^4, ..., $2^{\tau/2}$, 2^{τ} time steps. For the case of T = 16 (τ = 4) the sequence of initialization problems would be: 2, 4 and 16 time steps. The size of Δt in each case would be 8 t, 4 t respectively, and the number of steps required for the solution would be 1, 2 and 4 respectively, for a total of 7. This number still compares favorably with 16. In fact, for large values of T, the number of initializations required according to the above process is $\log_2 \log_2 T$. It is easy to verify that under these conditions the ultimate performance of the algorithms for large T is still of order $\log_2 T$.

The results of the solution of the nonlinear problem are outlined in Table II for a variety of intervals, time step sizes and initialization methods. Both the serial and parallel methods yield the exact same final result. Differences do exist, of course, when t is changed particularly as Δt becomes large. Also, the number of serial steps required does depend on the method used and the initialization scheme used. The multi-step ini-

Table II. Serial vs. Parallel Trapezoidal Integration, Nonlinear Example

			Serial		Parallel		
Δt	τ	t_{max}	# steps	Avg iter/step	steps/iter	#iter (flat)	# iter (seq)
.1	4	1.6	16	4	4	3	2
.1	5	3.2	32	3.91	5	4	3
.1	6	6.4	64	3.81	6	4	3
.2	4	3.2	16	4.88	4	4	3
.2	5	6.4	32	4.69	5	4	3
.4	4	6.4	16	7.75	4	4	4
.8	3	6.4	8	*	3	4	4

(*) No convergence achieved.

tialization method used involves a preliminary flat start solution with a larger time step. The interpolated results for this solution are then used to initiate the solution with the desired time step Δt.

A number of rather interesting observations can be made from Table II. First, it can be seen that parallel solutions indeed result in fewer total steps, as expected. More important, however, is the observation that the total number of iterations required for parallel solutions remains small. For flat start it is comparable to serial solutions, and for sequential starts it is consistently lower, as expected. Furthermore, some cases that run into convergence difficulties for traditional serial trapezoidal integration (those involving very large Δt) did not show that tendency for parallel solutions. Which might point toward a potential unexpected advantage of parallel computation: more reliable convergence characteristics. Of course, before such a statement can be made, a more thorough study of its convergence characteristics in a large class of problems must be performed.

c) TPBVP Example

Consider the following simple TPBVP:

$$\begin{aligned} \dot{x}^1 &= -2.8\, x^1 - 1.8\, x^2 + t \qquad & x^1(0) &= 0 \\ \dot{x}^2 &= -7.2\, x^1 - 8.2\, x^2 + 1 \qquad & x^2(.4) &= 0 \end{aligned} \tag{25}$$

This and any similar problem can be solved either by shooting methods or by direct methods [23]. The shooting methods are iterative and are of no interest here. The direct methods correspond to the formulation described in the previous section. All T equations for all T time steps are treted simultaneously. Figure 7 illustrates the topology of $\underline{A}$ after this formulation with Δt=.05 and Figure 8 illustrates the same topology after the odd-even permutation and fill-in.

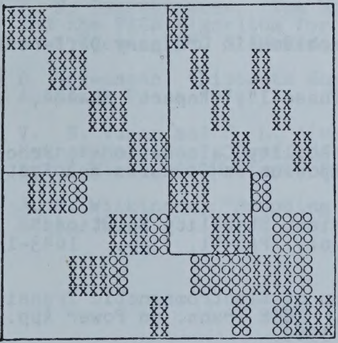


Fig. 8: $\underline{A}$ for TPBVP after permutation and fill-in.

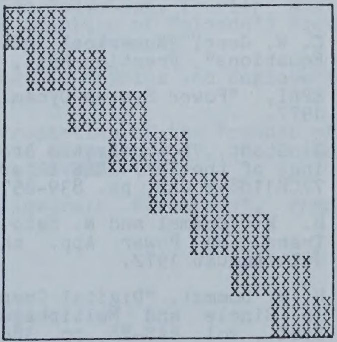


Fig. 7: Detailed topology of $\underline{A}$ for TPBVP example.

SPARSITY AND NUMERICAL ACCURACY

There are some limitations of the algorithm that deserve mention. The first one is with regard to sparsity preservation. The matrix blocks d, u and , so neatly represented by single symbols, are in reality n x n matrices, and may in general be sparse. Furthermore, the generation of the fills requires the product of of these matrices. It is well known that the product of sparse matrices is in general a less sparse matrix [24]. Furthermore, for connected networks there exists some n' < n, such that the product of n' matrices with the same topology yields a full matrix. It is therefore likely that the use of the recursive re-ordering algorithm will require the manipulation of full n x n matrices. Some situations when this limitation is not serious are:

a) Oftentimes the system is of sufficiently low order (say 20th or 30th) due to reduction that it is likely that all the matrices involved are full anyway. On-line applications frequently require that extensive reduction on the original system be performed. In these cases there are absolutely no sparsity disadvantages in the proposed algorithm.

b) It is always possible to deal efficiently with full nxn matrices, even for large n. In the days of strictly serial processors this would have sounded totally unacceptable. However, it is in fact possible to perform the required products in time of order n by utilizing n^2 independent processors, and even further gains are possible if either pipelining or array processors are used. Admittedly, the expense is likely to be quite high, but so are the potential rewards for those selected key applications that do warrant it.

A second potential problem is numerical error accumulation. It is well known that in order to minimize the accumulation of numerical errors during the solution it is necessary to perform the pivoting in a certain order [25]. During most power system matrix manipulations the problem of pivoting for numerical error minimization is largely ignored in favor of fill reduction due to the usually well behaved diagonal dominant nature of the matrices involved; the fact that sparsity is preserved also aids in the reduction of error accumulation. The numerical performance of the proposed method is not immediately obvious. One thing should be clear, however. As already mentioned, once a solution has converged, the results are just as reliable as those of serial trapezoidal integration.

CONCLUSIONS

An algorithm for the parallel solution of initial and multi-point boundary value problems has been presented. The algorithm is based on the numerically stable trapezoidal integration rule. It accomplishes parallelism between time different time steps. It is expected to be most efficient for full and block diagonal sets of equations. Further work on the convergence characteristics of the method in the case of nonlinear problems and on the possible numerical problems due to pivoting restrictions deserve further consideration.

ACKNOWLEDGEMENT

This research has been funded by NSF Grant ENG76-12254.

REFERENCES

[1] C. W. Gear, "Numerical Initial Value Problems in Ordinary Differential Equations", Prentice Hall, 1971.

[2] EPRI, "Power System Dynamic Analysis-Phase I", Report EL-484, July 1977.

[3] B. Stott, "Power System Step-by-Step Stability Calculations", Proceedings of the 1977 IEEE International Symposium on Circuits and Systems, 77CH1188-2 CAS, pp. 839-857.

[4] H. W. Dommel and N. Sato, "Fast Transient Stability Solutions", IEEE Trans. on Power App. and Syst., vol. PAS-91, pp. 1643-1650, July/August 1972.

[5] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single and Multiphase Networks", IEEE Trans. on Power App. and Syst., vol. PAS-88, pp. 388-399, April 1969.

[6] H. W. Dommel and W. S. Meyer "Computation of Elecromagnetic Transients", Proc. IEEE, Vol. 62, pp. 983-993, July 1974.

[7] EPRI EL-566-SR, "Exploring Applications of Parallel Processing to Power System Analysis Problems", October 4-7, 1977, Palo Alto, Ca.

[8] P. H. Enslow, Jr. (editor, "Proceedings of the 1976 International Conference on Parallel Processing", IEEE Publication 76CH1127-OC, August 24-27, 1976, Wayne State University, Detroit, Mich.

[9] G. Goos and J. Hartmanis (editors), "Parallel Processing", Springer-Verlag 1975, Proceedings of the Sagamore Computer Conference, August 20-23, 1974.

[10] J. F. Traub (editor), "Complexity of Sequential and Parallel Numerical Algorithms", Academic Press, 1973, Proceedings of the Symposium on Sequential and Parallel Numerical Algorithms, Carnegie Mellon, Pittsburgh, Pa., May 16-18, 1973.

[11] P. H. Enslow, Jr. (editor), "Multiprocessors and Parallel Processing", John Wiley, 1974.

[12] O. Wing, "Parallel Solution of a Sparse System of Linear Equations", Proceedings of [7], pp. 137-146.

[13] J. Fong and C. Pottle, "Parallel Processing of Power System Analysis Problems Via Simple Parallel Microcomputer Structures", Proceedings of [7], pp. 265-285.

[14] W. L. Hatcher, F. M. Brasch, Jr., and J. E. Van Ness, "A Feasibility Study for the Solution of Transient Stability Problems by Multiprocessor Structures", IEEE PES Winter Meeting, New York, New York, Jan. 30-Feb. 4, 1977.

[15] F. L. Alvarado, "On the Implementation of Parallel Computation by Network Diakoptics", Proceedings of [7], pp. 209-222.

[16] F. L. Alvarado, "Computational Complexity in Power Systems", IEEE Trans. on Power App. and Syst. PAS-95, No. 4, July/August 1976.

[17] W. L. Miranker and W. M. Lininger, "Parallel Methods for the Numerical Integration of Ordinary Differential Equations", Mathematics of Computation, Vol. 21, 1967, pp. 303-320.

[18] P. B. Worland, "Parallel Methods fo the Numerical Solution of Ordinary Differential Equations", IEEE Transactions on Computers, Vol. C-25, October 1967, pp. 1045-1048.

[19] H. S. Stone, "An Efficient Parallel Algorithm for the Solution of a Tridiagonal System of Equations", Journal of the ACM, Vol. 20, No. 1, Jan. 1973, pp. 27-38.

[20] O. Buneman, "A Compact Non-iterative Poisson Solver", Rep. 294, Inst. for Plasma Res., Stanford U., 1969.

[21] B. L. Buzbee, G. H. Golub and C. W. Nielson, "On Direct Methods for Solving Poisson's Equations", SIAM J. Numer. Anal., Vol. 7, No. 4, Dec. 1970.

[22] P. N. Swarztrauber, "The Method of Cyclic Reduction, Fourier Analysis and the FACR Algorithm for the Discrete Solution of Poisson's Equation on a Rectangle", SIAM Review, Vol. 19, No. 3, July 1977.

[23] D. Greenspan, "Discrete Numerical Methods in Physics and Engineering", Academic Press 1974.

[24] V. S. Vasan and F. L. Alvarado "Some Properties of the Product of Two Sparse Matrices", Dept. of Electrical and Computer Eng., the University of Wisconsin-Madison, Rep. ECE-77-10, November 1977.

[25] J. H. Wilkinson, "Rounding Errors in Algebraic Processes", Prentice Hall, 1963.

A New Approach to Power System Reliability Calculations

G. T. Heydt J. B. Franklin
Purdue Electric Power Center
Purdue University
West Lafayette, IN 47907

Abstract

This paper discusses a new approach to the rapid evaluation of reliability at power system busses. The approach is a sensitivity technique somewhat analogous to power transmission distribution factors. The method is applicable to calculation of bus reliability for cases which are changed from a base case. Proposals for reliability design are given.

Introduction

The probabilistic nature of power systems has produced interest on the part of system planners and operators which has grown with the complexity of power systems. Earlier papers often considered the general concepts of system reliability and the utilization of the average rate techniques [1,2]. Another extensive and useful study resulted in an industry-oriented computer program from frequency and duration methods of calculating system reliability [3,4]. The usefulness of Markov process theory in reliability engineering was shown in other areas of expertise [5] and was used in the context of power systems by Billington [6]. However, the computational requirements of Markov process methods are so extensive that these techniques are difficult to apply in practical problems. Sensitivity methods, on the other hand, are frequently used in engineering to obtain solutions quickly although, perhaps, at reduced accuracy. Examples of sensitivity-type solutions occur in electric power engineering in power transmission distribution factor techniques for fast power flow studies [7-9], and fast structural analysis of transmission towers [10]. Franklin [11,12] has proposed a sensitivity factor relating bus reliability to transmission system component reliability. Applications of this sensitivity factor include the design of transmission system reliability which results in given system performance. The technique is discussed in this paper.

Mathematical Basis

A model which has been used to describe repairable systems is that of a recursive Markov process [6]. A Markov process can be used in describing systems where failure rates and repair rates are constant and the system is "memory-less" (i.e., the conditional probabilities of going from one state to the next are not dependent upon any past system state.) This is a reasonably accurate description of an operating power system. The initial condition of a system described by a Markov process determines the probability of being in any state after a given number of steps. These probabilities are functions of the initial condition and the number of steps that have occurred. Considering the case of a single repairable component, the following definitions are made:

$P_o(t)$ = probability that the component is operable at time t

$P_1(t)$ = probability that the component is failed at time t

λ = failure rate

$\mu = \frac{1}{r}$ = repair rate (r = repair time).

Assume that during an incremental time interval, dt, the joint probability of two events occurring is negligible. Then,

$$P_o(t+dt) = P_o(t)(1-\lambda dt) + P_1(t)(\mu dt)$$

$$P_1(t+dt) = P_o(t)(\lambda dt) + P_1(t)(1-\mu dt).$$

From these equations,

$$\frac{P_o(t+dt)-P_o(t)}{dt} = -\lambda P_o(t) + \mu P_1(t).$$

In the limit, as dt approaches zero, in matrix form,

$$\begin{bmatrix} P_o'(t) \\ P_1'(t) \end{bmatrix} = \begin{bmatrix} -\lambda & \mu \\ \lambda & -\mu \end{bmatrix} \begin{bmatrix} P_o(t) \\ P_1(t) \end{bmatrix}.$$

The coefficient matrix of λ and μ terms is called the Stochastic Transitional Probability (STP) matrix. The equations are linearly dependent; to find steady state solutions, note that $P_{o\ ss}$ and $P_{1\ ss}$ sum to unity,

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\lambda & \mu \\ 1 & 1 \end{bmatrix} \begin{bmatrix} P_{o\ ss} \\ P_{1\ ss} \end{bmatrix},$$

which yields upon solution,

$$P_{o\ ss} = \frac{\mu}{\lambda+\mu} \quad \text{and} \quad P_{1\ ss} = \frac{\lambda}{\lambda+\mu}.$$

This concept can be applied to a system of N components where there are 2^N states of the system. These computations involve a very large number of equations for practical power systems. However, savings are realized due to the extreme sparsity of the STP matrix. Also, it is not necessary to find state probabilities where a large number of components have failed as these probabilities are negligible in practical systems. In order to preserve the original STP matrix, the system of equations is solved by the conjugate gradient method for linear equations [13]. This technique is based on the Gram-Schmidt orthogonalization process and is an iterative algorithm.

Binary Bus Connection Matrix

Having found the required probabilities of the system states, it is necessary to determine which busses of the system are in service during each system state. This can be done efficiently using the binary bus connection matrix [7]. The binary bus connection matrix is a binary (or logical) matrix whose ijth element is defined as $(B)_{ij}$,

$$(B)_{ij} = \begin{cases} \text{1 (or TRUE) if bus i is connected to bus j} \\ \qquad \text{through one line or if i=j} \\ \text{0 (or FALSE) otherwise.} \end{cases}$$

The operation of "logical multiplication" can be performed between two of these matrices and is the same as normal matrix multiplication except that algebraic multiplication is replaced by the logical "AND" and addition is replace by the logical "OR". The result of multiplying a binary bus connection matrix by itself yields a matrix which has for its ijth entry $(B^2)_{ij}$. Higher "powers" of the B matrix yield B^m which has the property,

$$(B^m)_{ij} = \begin{cases} 1 \text{ if bus i and bus j are connected} \\ \quad \text{through m lines or less} \\ 0 \text{ otherwise.} \end{cases}$$

By raising the matrix to successive powers, it is possible to determine whether a certain bus is connected to the transmission system grid.

In the analysis presented in this paper, the B matrix is formed for each state of the system. Those buses which are connected to the transmission grid are determined for each state. Using the known probability of occurrence of each state, the total bus reliability is evaluated.

Sensitivity of Bus Reliability with Respect to r, λ

The method outlined above yields the reliability of bus i, T_i, for given r and λ. T is a vector valued function of vectors R and Λ containing component repair times, r_j, and failure rates, λ_j, respectively. Consider a Reliability Sensitivity Matrix (RSM) of T with respect to R,

$$(RSM_R)_{ij} = \frac{\partial T_i}{\partial R_j} = \frac{\partial T_i}{\partial r_j}$$

and the RSM with respect to Λ,

$$(RSM_\Lambda)_{ij} = \frac{\partial T_i}{\partial \Lambda_j} = \frac{\partial T_i}{\partial \lambda_j}.$$

These RSM's are the Jacobian of T with respect to R and Λ. The RSM's are calculated using

$$(RSM_R)_{ij} \approx \frac{T_i^b - T_i^a}{r_j^b - r_j^a}$$

$$(RSM_\Lambda)_{ij} = \frac{T_i^b - T_i^a}{\lambda_j^b - \lambda_j^a}$$

where T_i is the probability of bus i being in service, r_j is the repair time of component j, λ_j is the failure rate of component j, and the subscripts a and b refer to a base case and changed case.

The RSMs are used to calculate T in changed cases (such as case "c"),

$$T^c = T^a + RSM_R\ (R^c - R^a) + RSM_\Lambda\ (\Lambda^c - \Lambda^a).$$

Thus, a system with bus reliability T^a may be re-designed to a desired reliability T^c by varying R and/or Λ.

Illustrative Example

Figure 1 depicts a five bus sample system. Table 1 gives the salient statistical information for this system. The RSMs for this system were calculated as described above; to conserve space, only a submatrix of RSM_Λ is shown here,

$$RSM_\Lambda = \begin{array}{c} \\ \text{bus2} \\ \text{bus3} \\ \text{bus4} \\ \text{bus5} \end{array} \left[\begin{array}{ll} \text{line 2} & \text{line 3} \\ -0.00038712 & 0 \\ -0.00038223 & -1.56005541 \\ -0.00038462 & 0 \\ -0.00011793 & 0 \end{array} \right]$$

This matrix is used to predict T at each system bus for changes of λ_2 and λ_3,

$$T^c = T^b + \left[RSM_\Lambda\right] \begin{bmatrix} \Delta\lambda_2 \\ \Delta\lambda_3 \end{bmatrix}.$$

In Table 2, the changed case reliabilities, T^c, so calculated are compared to T^c calculated exactly.

Table 1

Line	Failure rate, $\lambda(\text{time}^{-1})$	Repair time, r(time)
1	0.006	1.70
2	0.007	1.40
3	0.008	1.60
4	0.005	1.30
5	0.011	1.80
6	0.007	1.25

Table 2

Bus	T_c linear method	T^c "exact"	$-T^c$ linear method	$1-^c$ "exact"
2	0.9999973801	0.9999973801	0.0000026199	0.0000026199
3	0.9876722568	0.9876712430	0.0123277432	0.0123287570
4	0.9935393742	0.9935393742	0.0064606258	0.0064606258
5	0.9998307893	0.9998307893	0.0001692107	0.0001692107

It is concluded from this brief example that it is possible to use the principle of superposition in the above equation for T^c to rapidly calculate changed case reliability using sensitivity factors. Other tests, not shown here, indicate that changes of r and/or λ of about 20% continue to show linear behavior of T versus R and Λ.

Conclusions

A sensitivity method has been demonstrated in this paper by which bus reliability may be calculated for changed repair time and failure rate. Alternately, the method permits the calculation of required repair time or failure rate improvement to produce a given bus reliability improvement. The method is a linearized method, but accuracy is typically excellent when R or Λ entries vary by less than 20%. This method is noval in that it permits design of reliability levels.

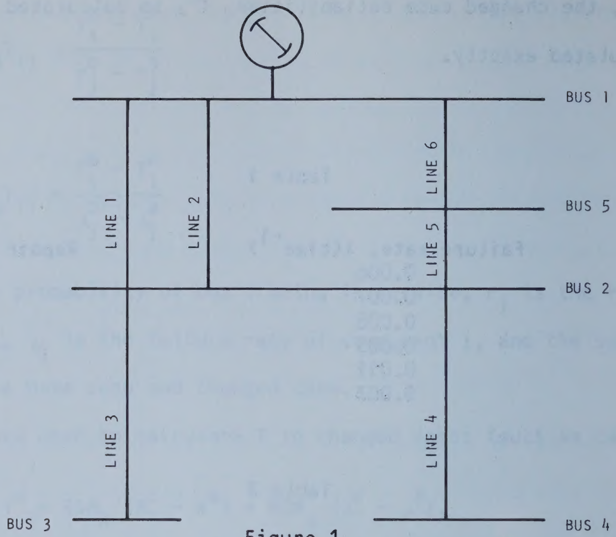


Figure 1

Five Bus Example

References

[1] "A Review of Some Basic characteristics of Probability Methods as Related to Power System Problems", C. W. Watchorn. *IEEE Trans. on PAS*, Vol. 83, July 1974, pp. 737-743.

[2] "A Probability Method for Transmission and distribution Outage Calculations", Z. G. Todd. *IEEE Trans. on PAS*, Vol. 83, July 1964, pp. 695-701.

[3] "Power System Reliability I - Measure of Reliability and Methods of Calculation", D. P. Gaver, R. E. Montmeat, A. D. Patton. *IEEE Trans. on PAS*, Vol. 83, July 1964, pp. 727-737.

[4] "Power System Reliability II - Applications and a Computer Program", F. E. Montmeat, A. D. Patton, J. Zemkoski, D. J. Cumming. *IEEE Trans. on PAS*, Vol. 84, July 1965, pp. 636-643.

[5] *System Reliability Engineering*, G. H. Sandler. Prentice-Hall, Inc., 1963.

[6] Power System Reliability Evaluation, R. Billington. Gordon and Brench Science Publishers, Inc., 1970.

[7] EE 532 course notes, Purdue University, 1978 by G. T. Heydt.

[8] H. E. Brown, "Solutions of Large Networks by Matrix Methods", Wiley, 1975, Chapter 7.

[9] W. M. Grady, G. T. Heydt, "Fast three Phase Load Flow Studies", Purdue University Technical Report, TR-EE 72-28, December, 1972.

[10] "Rapid Methods for Transmission Tower Structural Analysis and Design", G. Heydt. W. Grady, IEEE Trans. on PAS, July/August 1975, pp. 1223-1231.

[11] "A Preliminary Report on A Sensitivity Technique for the Calculation of Power System Reliability", J. B. Franklin, Purdue University Technical Report, PCTR-59-78, May 1973, West Lafayette, IN 47907.

[12] "Power System Reliability Calculations Employing a Sensitivity Technique", J. B. Franklin, G. T. Heydt, 1978 Midwest Power Symposium, Lincoln, Nebr., October 1978.

[13] Mathematical Methods for Digital Computers, Ralston and Wilf., Vol. 1.

CONTROL CONCEPTS FOR BULK INTERCONNECTED POWER SYSTEMS IN AN EMERGENCY STATE

JEROME MEISEL
Department of Electrical and Computer Engineering
Wayne State University
Detroit, Michigan 48202

ABSTRACT

This paper presents a summary of recent research into Emergency State control of bulk interconnected electric power systems. The purpose of this control is to augment transient stability margins following a major disturbance to the system. The nature of this problem is first outlined, followed by a description of the solution concepts developed in this project. These concepts involve (i) the definition of a localized control space, (ii) the use of an optimal-aiming strategy algorithm for making control decisions, and (iii) implementation considerations particularly through switched passive network elements. References to other publications giving a more extensive discussion into each of these problem areas are included.

INTRODUCTION

Present day electric power systems contain large scale generating plants interconnected to load centers by long distance high voltage transmission lines of high capacity. The electrical capacity of an a-c transmission line is directly proportional to the square of the voltage on the line and inversely proportional to the reactive impedance of the line. In order to achieve higher transmission capacities, engineers over the years have designed lines operating at ever increasing voltages, with some lines in the United States now operating as high as 765 kV. The technology of high voltage a-c transmission has progressed to the stage where thousands of megawatts of electric power are transmitted over single corridors. Thus a transmission line, based on its rated voltage and reactive impedance, has a set maximum capacity much in the same manner that a three lane-per-side freeway has a designed maximum throughput for automobile traffic.

Normally engineers operating a power system want to be able to operate transmission facilities at levels up to their designed limit. However, a dynamic problem does not allow full capacity to be achieved in practice. On the occurrence of a significant system disturbance, such as a major fault or a major loss of load, the system moves from a Normal or Alert State into an Emergency State requiring control action to limit and contain unsuitable operating conditions [1]. After protective switchgear has acted to prevent severe component damage, an additional problem, known as the transient stability problem, is encountered due to the severely perturbed condition of the system. This transient stability problem refers to the maintenance of generator synchronization following the significant system disturbance which has moved the system into an Emergency State. For a given disturbance, the faster it is removed by switchgear action, the less will be its effect on the system. For a severe disturbance that is not cleared away fast enough, the shock to the system may be so large that loss of synchronism occurs and the disturbance thereby propagates causing a cascading failure. The famous U.S. Northeast blackout on November 9, 1965 began with the loss of a transmission corridor, thus isolating a significant amount of generation from its load. Insufficient corrective actions occurred and the situation cascaded into the subsequent disaster.

Usually disturbances, in the form of faults, are removed by switchgear action. The maximum allowable time available for the switchgear to remove a certain fault is a function of the power being transmitted over the transmission lines. A greater level of power transmission requires a given fault to be cleared in a shorter time. So if we have installed switchgear with fixed operating times, the maximum allowable power transmission for which faults can be cleared without loss of synchronism is thus limited.

In summary, we have two limits on the allowable transmission capacity for a transmission line. These are as follows:

1. The designed capacity, which is related to the voltage level and the line reactance.
2. The capacity which is related to the speed at which protective switchgear can clear disturbances.

The need for the Emergency State controls arises because modern interconnected power system transmission lines are being limited by the second limit and not the first.

Switchgear manufacturers have markedly reduced the time required for their equipment to isolate and clear distrubances. However, the capacity of transmission facilities has been increasing thus requiring physically larger switchgear which have correspondingly longer operating times. In addition increased size of generator units, designed with lower per-unit inertia parameter values, and longer lengths of transmission lines have both contributed to shortening the allowable disturbance clearing times. The sole reliance on the speed of switchgear to clear disturbances so that synchronization is maintained seems to be no longer possible. The purpose of our research efforts is to develop and test practical control algorithms that, when employed with auxiliary apparatus, allow active, on-line, computerized control to insure synchronous or stable operation of interconnected power systems following a major perturbation, thus aiding the transition from an Emergency State back to an Alert and Normal State.

The present industry practice is to rely on fast fault-clearing times and careful relay coordination to achieve free-running stable system trajectories. These arrangements are extensively studied off-line against a multitude of system contingencies. This present research is suggesting an active feedback-type control during contingencies which may not be successfully handled without this control. As the bulk system grows, some of the inadequacies of the present industry practice are appearing in the western-doughnut, in the east-west interties, and in some of the cascading system failures in the eastern system. Techniques such as load shedding or load addition, reactance switching, fast valving, forced field excitation, etc., all come under the heading of Emergency State control means. Numerous investigators [2,3,4] have used these controls and shown their benefit. Our work has been aimed at developing a control strategy which would use these control means based on consideration of the nonlinear, interconnected multi-machine bulk system.

Our research into the Emergency State control problem has generally partitioned the problem along the following major lines of inquiry:

1. Defining a suitable localized control space requiring only locally available information.
2. Development of a feedback control algorithm capable of producing control decisions in the appropriate time frame using a nonlinear internal adaptive model.
3. Implementation of the control decisions through a variety of suitable physical means, generally using a hierarchical information processing structure.

This present paper describes each of these subareas together with our current concepts and results.

THE CONTROL STATE SPACE

Interconnected electric power systems cover a large geographic area. In such a system the state variables involved in the coupling forces between the various machines are generally the respective terminal voltages, torque angles, and torque angular velocities. If our state vector and corresponding state space for the system involves all of the machine angles and angular velocities, as is common in off-line studies, then we are faced with a crucial problem in on-line control. This problem consists of establishing a reference frame for these angles and angular velocity variables. For off-line simulation purposes three reference frames are commonly used. These are denoted as [5]:

1. The synchronous frame (SF)
2. The machine-one frame (MOF)
3. The center-of-angle frame (CAF)

All three of these frames have one major shortcoming if used for on-line control. This shortcoming involves the requirement that information be transferred globally or system-wide in order to determine values for the state variables. The requirement that measurements be telemetered around from bus to bus is considered to be not feasible in the time frame of practical consideration. Eventually perhaps some global influence may in the future be feasible as an over-ride or a higher level modifier on some control strategy [6], but at present our investigations are centered upon strategies making decisions based only on locally available information.

For control purposes we prefer a state space such that the present location of each machine or subsystem can be determined entirely from local measurements. Recent work by Zaborszky, Subramanian, et. al. at Washington University has been directed along these lines [7,8]. This new state space or reference frame allows for the location of the state of each local subsystem with respect to a local dynamic equilibrium state. This local dynamic equilibrium state is a state whereby the subsystem would be in equilibrium with all of its neighboring, or directly connected subsystems, if these neighbors stayed in their present state. Figure 1 illustrates the basic idea of this new localized reference frame termed a local equilibrium frame (LEF). Consider our local subsystem of interest to be connected to m-neighboring subsystems by transmission lines as shown in Fig. 1. Measurements of voltage and complex power are made at the local end of each transmission line. Based on these measurements, with some internally available information about the local subsystem, we can determine the state of the local subsystem with respect to an equilibrium state whereby the local subsystem would be in equilibrium with its m-neighbors if the m-neighbors remained stationary. Of course the m-neighbors are not stationary. However, the tracking of this local dynamic equilibrium state is proposed as a reasonable localized decision, which if accomplished globally, leads to a stable global equilibrium state.

The usual formulation of the LEF, as in [5,7,8], uses torque angle and angular velocity as state variables. Recent work indicates some advantage to using real and reactive power quantities in place of torque angles. This alternate localized state space, denoted as the (P-Q-ω) space, is presented in reference [9]. Instead of dealing with the LEF angle we here deal with the more directly measurable quantities of internal real and reactive power out of each machine. This latter space has all of the localized properties of the original LEF formulations.

The state equations for each local machine or coherent group of machines in the (P-Q-ω) space for an n-machine system are*

$$\dot{x}_i(t) = f_i(x_i(t)) + B_i u_i(t) \tag{1}$$

---

*Note: The term machine or single machine is used to also denote a coherent group of machines which dynamically move as an equivalent single unit.

with $x_i^T = (P_i, Q_i, \omega_i)$

$$f_i(x_i(t)) = \begin{bmatrix} -(Q_i(t) + B_{ii}E_i(t)^2)\omega_i(t) \\ (P_i(t) - G_{ii}E_i(t)^2)\omega_i(t) \\ \frac{1}{M_i}(P_{mi} - D_i\omega_i(t) - P_i(t)) \end{bmatrix}$$

$$B_i^T = (0, 0, \frac{1}{M_i})$$

for all $i = 1, \cdots, n$

Here ω_i is the angular velocity of machine i with respect to a synchronous reference frame. Equation (1) represents the local control model in the (P-Q-ω) space and can be used as an internal model for the control algorithm. The key factor in this (P-Q-ω) control model is that the current state at time t, namely $x_i^T(t) = (P_i(t), Q_i(t), \omega_i(t))$ can be determined at machine i from measurements available at the terminals of machine i.

In this space the local stable equilibrium state is defined by $\bar{x}_i^T(t) = (\bar{P}_i(t), \bar{Q}_i(t), \bar{\omega}_i(t))$, and can be found from

$$\bar{P}_i(t) = P_{mi} \tag{2}$$

$$\bar{Q}_i(t) + B_{ii}E_i(t)^2 = -[(E_i(t)I_{io}(t))^2 - (\bar{P}_i(t) - G_{ii}E_i(t))^2]^{1/2} \tag{3}$$

$$\bar{\omega}_i(t) = 0 \tag{4}$$

We therefore have the standard regulator control problem whereby the current local state $x_i(t)$ is obtained from system measurements and we desire to direct the system towards the current estimate of the local stable equilibrium state $\bar{x}_i(t)$ subject to the state equations (1). Practical limits on the control force $u_i(t)$ must also be included.

THE CONTROL ALGORITHMS

The state equations given by equation (1) including an additive control function u are of the general form

$$\dot{x}(t) = f(x(t),t) + B(x(t),t)u(t) \tag{5}$$

where our state function x maps a time interval $T = [t_o, t_1]$ into R^n, and control function u maps T into R^m. We further assume the control values to be limited by practical considerations such that

$$u(t) \in \Omega(x(t),t) \subseteq R^m \tag{6}$$

Our problem is to find a control function which drives the system from some initial state $x(t_o)$ to the state-space origin in some optimal fashion subject to the state equation (5) and the control constraints given by (6).

The classical optimal-control formulation of this problem generally proceeds by defining a cost functional in the form

$$J = \int_{t_o}^{t_1} L(x(t), u(t), t)dt \tag{7}$$

This functional maps the state and control functions over the interval T into a scalar real value J, subject to (5). By appropriately selecting this cost functional, we can reflect our desire to drive the system to the origin, and also to exclude impractical control values. The search for the optimal control function u* which yields the minimum value for J

"... involves infinite-dimensional function space optimizations which are generally impossible to solve analytically and approximation techniques which are so laborious and time consuming as to be virtually useless." [10]

The classical approach to this nonlinear regulator problem yields solutions in the form of open-loop control functions defined over the entire time interval, which are very dependent on the authenticity of the state equations relative to the real system being controlled. So even if solutions can be obtained, they are not very useful.

The optimal-aim strategies (OAS) developed by Barnard are designed to reduce infinite-dimensional state-space optimizations to finite-dimensional optimizations, which yield values of the control function in terms of the current time, state of the system, and present control function constraints [10-14]. The controls obtained are feedback in form, and are thus more immune to errors between the state-space model used internally by the algorithm and the real system being controlled. In fact, the control values are obtained in a closed analytic form implementable by continuous analog (i.e. non-sampling) hardware for the two dimensional regulator problem in the LEF or (P-Q-ω) space [9].

The OAS can be briefly summarized as follows. Consider a system described by $\dot{x}_i(t) = f_i(x_i(t))+B_i u_i(t)$, as in equation (1). Assume that a stable equilibrium state for this system, at time t, exists at $\bar{x}_i(t)$. The system is not presently in the equilibrium state, and may be heading for instability. We desire to use the control u_i to drive the system towards the equilibrium state. Referring to Fig. 2, the idea of the OAS is to use the control to point the state derivative vector $\zeta_i \epsilon \Lambda_i$ towards the aim state $x_{oi}(t)$, which is an equilibrium or any other appropriately chosen point. The reference vector $\rho_i(t)$ is defined as the vector from $x_i(t)$ to $x_{oi}(t)$. This reference vector indicates the direction we would like to give to ζ_i at time t. However, the structure of f_i and the range of control physically available may be such that this objective is unreachable. The OAS then selects among the set of feasible ζ_i the "closest," in some sense, to $\rho_i(t)$ at each time t. The definition of closeness gives rise to different versions of the OAS, and an optimal derivative vector denoted simply as $\dot{x}_i(t)$.

This selection is analytically formulated in terms of cost function minimizations. The various versions of the OAS algorithm are enumerated as follows:

1. Minimum angle version [9-11,19]:

$$J_i(\dot{x}_i(t)) \geq J_i(\zeta_i) \qquad \forall \zeta_i \epsilon \Lambda_i \tag{8}$$

$$\text{where } J_i(\zeta_i) = \frac{<\zeta_i, \rho_i>}{|\zeta_i| \cdot |\rho_i|}$$

2. Minimum norm version [12,18,19]:

$$J_i(\dot{x}_i(t)) \leq J_i(\zeta_i) \qquad \forall \zeta_i \epsilon \Lambda_i \tag{9}$$

$$\text{where } J_i(\zeta_i) = |\rho_i - \zeta_i|$$

3. Optimal proximity version [13]:

$$J_i(\dot{x}_i(t)) \leq J_i(\zeta_i) \qquad \forall \zeta_i \epsilon \Lambda_i \tag{10}$$

$$\text{where } J_i(\zeta_i) = \frac{d}{dt}|\rho_i(t)|$$

4. Maximum product version [14]:

$$J_i(\dot{x}_i(t)) \geq J_i(\zeta_i) \qquad \forall \zeta_i \epsilon \Lambda_i \tag{11}$$

$$\text{where } J_i(\zeta_i) = <-\nabla|f_i(x_i(t))|^2, \zeta_i>$$

Through a special selection of the aim states certain of these versions of the OAS become equivalent. For example reference [15] shows compatibility between the minimum angle and norm versions.

Each version of the OAS has a simple geometric interpretation corresponding to aligning admissible controlled system motion with a desired reference direction. The local model provides a mapping between the control value $u_i(t)$ and the optimal derivative vector $\dot{x}_i(t)$ at each instant. The model can thus be of very low order, and its current state is continuously updated using actual observations of the behavior of the real system. In this sense, the algorithm becomes a feedback algorithm. As a state feedback control, it doesn't need a knowledge of the past or future history of the state or the control. This feature is very important for a local control using only local information. For such a control scheme, the other controls in the system appear to be random perturbations.

CONTROL DECISION IMPLEMENTATION

The OAS algorithm outlined in the above section selects desired values for control torques on the shaft of each machine, or coherent group of machines, in the system. A key and still somewhat open problem in Emergency State control is to implement these desired torques through appropriate power system hardware, here referred to as control means. These means must be able to achieve a required rate-of-change such that they are effective in the transient stability time frame, on the order of a second, following a major disturbance.

Forced field excitation, fast valving, d-c line modulation, and energy storage systems are means potentially available as devices to augment transient stability. Also devices such as braking resistor banks, and switched capacitors and inductors, line and load switching all tend to rapidly change the electrical power out of the system generators [2]. Control of the electrical power injected by a machine alters the motion of this machine. The control means described above all modulate this electrical power well within the time frame of the transient stability problem. However, the means involving some switched electrical element in the network have a faster time response than the other means.

A two-level hierarchical control structure has been developed involving local controllers and area coordinators, as shown in Fig. 3, [16-18].* Here the local controller deals only with the needs of a single machine, or a coherent group of machines, which can be lumped into a single equivalent machine. These are indicated by the heavy dots in Fig. 3. The local controllers operate as parallel processors, and only communicate information to their area coordinator. The reverse communication from area coordinator to local controller is not required. The local controller deals with system measurements available locally at the single machine or its unit transformer. The job of each local controller is to decide upon the value of the electrical power out of its local machine such that the machine would be optimally directed from an Emergency State back towards an estimate of a stable equilibrium state or Normal State. This decision is made using the local equilibrium frame (LEF) [7-9]. Based upon this state point estimate in the LEF, the optimal aiming strategy (OAS) developed by Barnard is used to select a desired electrical power injection [9-14,18,19].

In general, the area coordinator is involved with a subarea of noncoherent machines and their local controllers geographically situated such that two communication requirements can be met:

1. The desired electrical control power injection values must be received by the area coordinator from each local controller.

---

*Note: The term local controller is being used to denote a local information processor, which passes on information to an area coordinator. The actuation of physical control means is under the direction of the area coordinator.

2. The area coordinator must have state measurement and system structure information and be able to send orders to the control means in the subarea under its jurisdiction.
The job of the area coordinator is to take the unconstrained demands from all of its local controllers, and decide how to allocate its available finite means to implement these demands according to some optimal policy.

We have performed a great many experiments on a representative test system consisting of 39 buses and 10 machines to show the usefulness of these concepts [19,20]. In many of these tests the area coordinator allowed the electrical power injections demanded by the local controllers to all be implemented within a continuous set of values with practical hard limits. This form of area controller in effect has sufficient means at its disposal to achieve all demands instantaneously within practical magnitude constraints. However, the practically significant problem with the coordinator design involves the analytic constraints imposed by physical hardware in the actual system.

The role of the coordinator responsible for n-noncoherent machines geographically located such that some information transfer is feasible can be summarized as follows [18]:

> To receive the n desired power values, ($u_i(t), i=1,\cdots,n$) from the n localized controllers, and dispatch values for the m available control means, ($\nu_i(t), i=1,\cdots,m$), which achieve these $u_i(t)$ values in some optimal manner.

A general relationship between the power control u and the control means ν in an area can be espressed as

$$u(t) = g(x(t),\nu(t),t) \tag{12}$$

with $\nu_i(t)\in[\underline{\nu}_i,\bar{\nu}_i]$, and $g: R^{3n}\times R^m\times R^1\to R^n$. The job of the area coordinator is to formulate the mapping g, and find the inverse mapping, in some optimal sense, such that given the value of u the corresponding value of ν is determined and implemented.

If these means are linearly related to the desired power control, then we can write

$$u(t) = C(x(t),t)\nu(t) \tag{13}$$

where C is a n×m matrix, which can be state and time dependent and here $m\leq n$. The matrix C is known by the coordinator, and depends on the means available and the structure of the system being coordinated. In computing the total electrical power u(t) according to the OAS, no limits are proposed. Thus in effect, $u_i(t)$ represents the electrical power out of the ith machine which the ith local controller demands to optimally aim the machine's trajectory without regard to what power is actually available through manipulation of the control means.

Now given the vector u(t), a least-square fit for the value of the means $\nu(t)$ is obtained through the generalized inverse of C in the form [c.f. 21]

$$\nu(t) = \sigma((C^TC)^{-1}C^Tu(t)) \tag{14}$$

where $\sigma(\cdot)=\mathrm{col}\ \sigma_i(w_i)$ and

$$\sigma_i(w_i) = \begin{cases} \bar{\nu}_i & , w_i\in(\bar{\nu}_i,\infty) \\ w_i & , w_i\in[\underline{\nu}_i,\bar{\nu}_i] \\ \underline{\nu}_i & , w_i\in(-\infty,\underline{\nu}_i) \end{cases}$$

Note that the saturation function is only applied to the computation of the means. Thus only one-way communication is needed from the local controllers to the area coordinator. Figure 4 illustrates a complete implementation of the n-local controllers feeding their demands to a single area coordinator. The area coordinator, with the assumption of (13), allocates the control means at its disposal to meet these demands. In general, a large interconnected power system would have a number of the area coordinators shown in Fig. 4.

Thus we have the above coordinator design available if a linear relationship does indeed exist between the value of the control means and the additive shaft power on each machine. As previously mentioned, a very desirable control means for the Emergency State control problem involves electrically switching resistive, capacitive, and possible inductive elements in the interconnecting network. Preliminary studies on this problec have centered on the attempt to see if dynamic braking resistors can be placed in a power system network such that they induce linearly related changes in generator electrical output power [22]. The results of this study using the 10-machine, 39-bus New England test system indicate that a reasonably linear relationship exists if:

1. Braking resistors are available on the generator terminal bus.
2. Shunt capacitors are also placed across these resistors to maintain their terminal voltage.
3. Series capacitors are placed between the generator terminals and the braking resistor to more closely couple the brake to the local machine.

Thus to date the coordinator design using switched passive network elements requires the configuration shown in Fig. 5 on each generator whose local controller requires control action. The coordinators can thus be incorporated into each local controller and placed on each machine, or coherent group of machines, provided that sufficient decoupling of control action is achieved.

SUMMARY AND CONCLUSIONS

This paper outlines design concepts intended to augment transient stability margins following a major disturbance to an interconnected power system which has caused the system to move into an Emergency State. These concepts have involved definition of a localized control space using locally available measurements. Through various versions of an optimal-aiming strategy algorithm, control decisions are made to maintain stable synchronous operation of the system generators. The crux of the problem that remains is to implement these control decisions through appropriate system hardware. Some preliminary results into this implementation problem are discussed, particularly into the use of dynamic braking resistors combined with suitably arranged series and shunt capacitors. Work is continuing on this implementation aspect of the control problem.

NOMENCLATURE LIST

x_i	local state vector
x_{oi}	local aim state
$\rho_i = x_{oi} - x_i$	local reference vector
u_i	local additive control
M_i	inertia constant for machine i
D_i	damping constant for machine i
P_{mi}	mechanical power input for machine i
P_i, Q_i	real and reactive power output for machine i, respectively
E_i	internal voltage of machine i
G_{ii}, B_{ii}	real and imaginary parts of the self admittance at the internal bus of machine i, respectively
I_{io}	equivalent current into bus i from all neighboring buses (see [9])
$\langle \cdot , \cdot \rangle$	denotes inner product
$\lvert \cdot \rvert$	denotes Euclidean norm

Superscript: - usually denotes equilibrium value

ACKNOWLEDGMENTS

This work was supported by the Division of Electric Energy Systems of the Department of Energy under contract no. E(49-18)-2074. The author would like to thank Mr. L. Fink, Dr. K. Carlsen, and Dr. T. Trygar for their comments and guidance throughout this work. Also the help of Professor A. Sen is gratefully acknowledged.

REFERENCES

1. L.H. Fink and K. Carlsen, IEEE Spectrum, Mar. 1978, pp. 48-53.
2. J. Meisel, 1975 Eng. Found. Conf.: Systems Engineering for Power-Status and Prospects, Henniker, N.H.
3. P.J. Donalek, 1976 Joint Power Gen. Conf., Buffalo, N.Y., Sept. 19-23, 1976.
4. R.H. Park, "Fast turbine valving," IEEE Trans. Power App. and Syst., vol. PAS-93, no. 2, Mar./Apr. 1974, pp. 1065-1073.
5. J. Meisel, 1977 JACC, San Francisco, Cal., pp. 747-754.
6. J. Zaborszky, Private correspondence, Mar. 1977.
7. J. Zaborszky, et al., Report No. SSM-7601-Parts I and II revised, Dept. of Systems Science and Mathematics, Washington University, St. Louis, Mo., Oct./Nov. 1976.
8. J. Zaborszky and A.K. Subramanian, 1977 JACC, San Francisco, Cal., pp. 737-746.
9. J. Meisel, PSCC VI, Darmstadt, Germany, Aug. 21-25, 1978.
10. R.D. Barnard, IEEE Trans. on Automatic Control, vol. AC-20, no. 2, Apr. 1975, pp. 200-208.
11. J. Meisel and R.D. Barnard, PSCC V, Cambridge, England, Sept. 1975, Paper 3.2/9.
12. R.D. Barnard, IEEE Trans. on Automatic Control, vol. AC-21, no. 3, June 1976, pp. 432-434.
13. R.D. Barnard, IEEE Int. Conf. on Circuits and Systems, N.Y., May 1978.
14. T. Potter, WSU-Power Systems Group report no. 78-11, Oct. 1978.
15. A. Sen and R.D. Barnard, WSU-Power Systems Group report no. 78-8, June 1978. (Also submitted to IEEE Trans. on Automatic Control.)
16. M.O. Mansur, Report No. TR-EE-75-34, Purdue University, School of Elec. Eng., Aug. 1975.
17. U.O. Aliyn, "Corrective Control for Transient Emergency State of Power Systems," Ph.D. Thesis, Purdue University, 1978.
18. J. Meisel, WSU-Power Systems Group report no. 78-1 (revised). (Also submitted to IEEE-PAS).
19. J. Meisel, Report No. 77-1, Eng. Found. Conf.: Systems Engineering for Power - Emergency Operating State Control, Henniker, N.H., 1977.
20. L. Elfstrom, et al., WSU-Power System Group report no. 78-4, Mar. 1978.
21. B. Noble Applied Linear Algebra, Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1969, Sec. 5.6.
22. M.L. Gilles, WSU-Power System Group report no. 79-12, Oct. 1978.

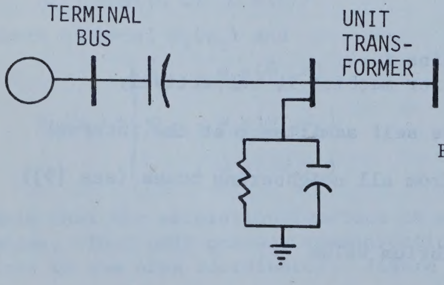
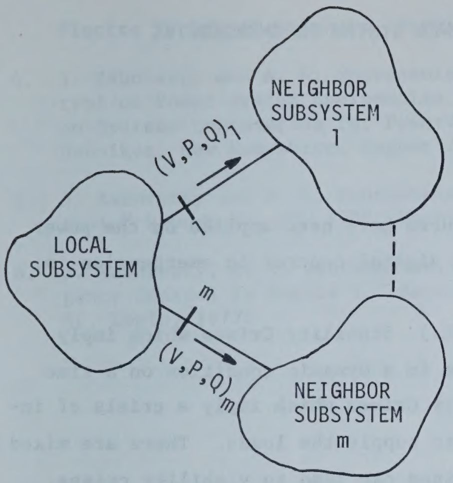


Fig. 5. Shunt and series capacitors required to isolate dynamic braking resistor control.


NEIGHBOR SUBSYSTEM 1
(V,P,Q)1
LOCAL SUBSYSTEM
m
(V,P,Q)m
NEIGHBOR SUBSYSTEM m

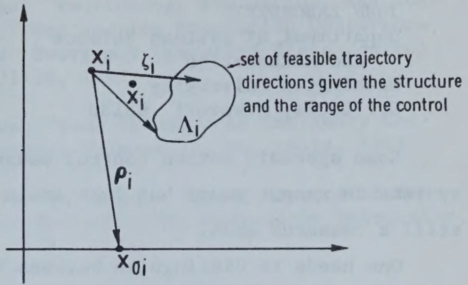

x_i
ζ_i
$\dot{x}_i$
set of feasible trajectory directions given the structure and the range of the control
Λ_i
ρ_i
x_{0i}

Fig. 1. Local subsystem connected to m-neighboring subsystems.

Fig. 2. Geometric description of the optimal aim strategy.

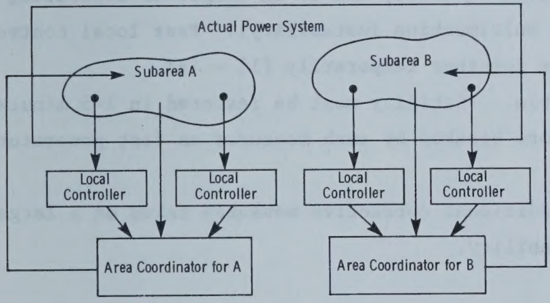

Actual Power System
Subarea A
Subarea B
Local Controller
Local Controller
Local Controller
Local Controller
Area Coordinator for A
Area Coordinator for B

Fig. 3. A two-level hierarchical control structure involving local controllers and subarea coordinators.

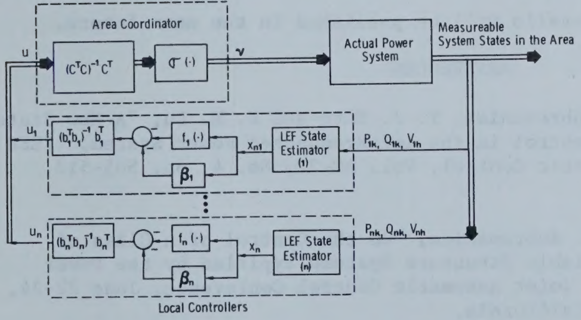

Area Coordinator
u
$(C^TC)^{-1}C^T$
$\sigma(\cdot)$
v
Actual Power System
Measureable System States in the Area
u_1
$(b_1^Tb_1)^{-1}b_1^T$
$f_1(\cdot)$
x_{n1}
LEF State Estimator (1)
P_{1k}, Q_{1k}, V_{1h}
β_1
u_n
$(b_n^Tb_n)^{-1}b_n^T$
$f_n(\cdot)$
x_n
LEF State Estimator (n)
P_{nk}, Q_{nk}, V_{nh}
β_n
Local Controllers

Fig. 4. Implementation of n-local controllers feeding unconstrained demands to a single linear area coordinator.

CONTROL OF THE LARGE INTERCONNECTED POWER SYSTEM IN EMERGENCIES

JOHN ZABORSZKY
Department of Systems Science
and Mathematics
Washington University
St. Louis, Missouri 63130

Some sporadic active control measures have been applied on the power systems in recent years but full scale digital control in emergencies is still a research area.

One needs to distinguish between 1.) Stability Crises which imply electromechanical breakup of the system in a dynamic condition on a time scale of a few seconds and 2.) Viability Crises which imply a crisis of insufficient generation or transmission to supply the loads. There are mixed types and, in particular, stability crises can lead to viability crises. In either situation five regimes of control can be distinguished [5] - [6].

1.) One unit is in danger of losing stability and leaving the system. This requires fast local measures to stop it [6].

2.) System is oscillating violently and is in danger of separating into islands (multiswing or multimachine instability). Fast local control is needed to hold the system together temporarily [1] - [4].

3.) System is not viable. Viability must be restored in 1-5 minutes in a provisional manner before breakup by such measures as fast generator runback or load dropping.

4.) Same as 3.) but additional corrective measures taken on a larger time scale for temporary viability.

5.) Restoration.

Solutions have been worked out for regimes 1,2,3 [1] - [6] mostly based on a "local equilibrium statespace" which is equivalent to the conventional statespace but allows control measures to be taken at the local bus [1] - [4]. Further details will be published in the near future.

REFERENCES

1. J. Zaborszky, A. K. Subramanian, T. J. Tarn and K. M. Lu, "A New State Space for Emergency Control in the Interconnected Power System," IEEE Transactions on Automatic Control, Vol. AC-22, No. 4, pp. 505-517, August 1977.

2. J. Zaborszky and A. K. Subramanian, "On the Control of a Class of Large, Nonlinear, Variable Structure Systems Typified by the Power System in Emergency," Joint Automatic Control Conference, June 22-24, 1977, San Francisco, California.

3. J. Zaborszky and A. K. Subramanian, "On Computer Control of the Large Interconnected Power System in Emergencies," Proceedings of the World

Electro Technical Congress, Section 7, Paper No. 30, Moscow, June 1977.

4. J. Zaborszky and A. K. Subramanian, "Monitoring, Evaluation, and Control of Power System Emergencies," Engineering Foundation Conference on Systems Engineering for Power: Emergency Operating State Control, Henniker, New Hampshire, August 21-26, 1977.

5. J. Zaborszky and A. K. Subramanian, "Some Thoughts on Emergency Control," Report No. SSM 7707, Washington University, St. Louis, 1977.

6. J. Zaborszky, A. K. Subramanian, K. V. Prasad, "Basic Aspects of Emergency Control in Regime I," Report No. SSM 7709, Washington University, St. Louis, 1977.

AREA LOAD-FREQUENCY CONTROL

T.M. Athay
Systems Control, Inc.
Palo Alto, California

H.G. Kwatny
Drexel University
Philadelphia, PA

R.G. Smith
Systems Control, Inc.
Palo Alto, California

1.0 INTRODUCTION

An advanced Automatic Generation Control (AGC) software package is currently being developed in a U.S. Department of Energy demonstration project. The scope of this project includes the design, implementation and testing of an integrated and coordinated set of algorithms for AGC. A major function in the overall structure is that of Load-Frequency Control (LFC) and this paper summarizes the current stage of design of the LFC algorithms. The emphasis of the paper is on a discussion of LFC objectives and on the use of some ideas from modern control theory to derive basic control structures which reflect those objectives.

2.0 AUTOMATIC GENERATION CONTROL

2.1 Conventional AGC Structure

Four basic objectives of power system operation during normal operating conditions can be associated with automatic generation control:

1. Matching total system generation to total system load
2. Regulating system electrical frequency error to zero
3. Distributing system generation amongst control areas so that net area tie flows match net area tie flow schedules
4. Distributing area generation amongst area generation sources so that area operating costs are minimized.

The first objective is conventionally associated with system primary or governor speed control; turbine speed governors respond proportionally to local frequency deviations and normally bring the rate-of-change of frequency to zero within a time-frame of several seconds. The latter three objectives are accomplished by supplementary controls directed from area control centers. The second and third AGC objectives are classically associated with the regulation function, or load-frequency control, while the fourth objective is associated with the economic dispatch function of automatic generation control. The latter two functions typically operate in a time frame from several seconds to several minutes.

Introduced nearly 30 years ago, the tie-line bias control strategy [1] is utilized in most interconnected power systems to accomplish the regulation function of AGC. In this approach the second and third objectives are combined and each area attempts to regulate its area control error (ACE) to zero, where

$$ACE \triangleq M - S^{o} + B\,(f - f^{o}) \qquad (1)$$

and

M = net tie flow out of the control area
S^{o}= scheduled tie flow out of the control area

B = area frequency bias constant (positive real)
f = actual frequency
f^o = scheduled frequency

A thorough discussion of the various aspects of tie-line bias control is provided in [1,2] and in other AGC literature. Very briefly, this strategy provides a steady-state target according to which each area meets its own load during normal conditions in the interconnection, contributes to frequency regulation and provides assistance to external areas when necessary. Although based on steady-state arguments, the tie-line bias control strategy has the important advantage that no inter-area communication is required, and it is on this basis that the decentralized automatic generation control of interconnected systems has been achieved.

The performance of conventional AGC has often been unsatisfactory however [3]. While some significant reasons for poor performance cannot be directly attributed to the control structure per se, not enough generating units on control for example, others can arise from the nature of the basic control algorithms as well as from the diverse data processing structures with which they are implemented. An important observation is that conventional AGC structures do not fully reflect the essential tracking nature of the problem. For example some are effectively pure integral controllers whose overall loop gain must be rather low to maintain stability. As a result tracking performance is limited, particularly during sustained, rapid load changes such as the morning load pickup. At such times manual operator intervention is generally required in order to maintain ACE within acceptable limits. Evidence of sustained fluctuations in frequency caused by inappropriate control has been reported [4]. In addition, a lack of coordination between the AGC regulation and economic dispatch functions can result in conflicting requirements on generating units and inefficient control.

2.2 Advanced Automatic Generation Control

The overall objective of this development and implementation project is to demonstrate that significant improvements in AGC performance can be obtained by better utilization of existing control center hardware capability and more extensive application of current control system knowledge. The improved AGC software package, which is currently in an early stage of development, will be implemented and demonstrated at the Wisconsin Electric Power Company (WE) Control Center. The WE energy control center is representative of the current state-of-the-art, utilizing a hierarchical system of dual CDC Cyber and quad CDC 1700 computers [5].

An integrated and coordinated set of algorithms is being developed for AGC. Three respective functions will be performed: short-term load prediction, dynamic optimal dispatch and load-frequency control. Before reporting on the current status of the prototype design structures for the LFC function, a brief description of the first two functions is provided.

A short-term load predictor produces a forecast of the secular component of the daily load curve at five minute intervals for 1-2 hours into the future. The dynamic optimal dispatch then determines unit trajectories which track the secular load so that production costs are minimized in a time-integrated manner, rather than on an instant-by-instant basis as in conventional economic dispatch. The dispatch algorithm is based on the combined use of a minimum marginal cost method and a dynamic programming successive approximations method. Among other things, the combination of short-term load forecasting and dynamic dispatch provides the capability for valve point loading (which will be demonstrated on two WE coal-fired units) and provides the plant operators with an advanced knowledge of future generation demand.

2.3 Load-Frequency Control

Over the last few years numerous researchers have explored certain aspects of the load-frequency control problem in the context of modern control theory. One of the earliest studies was that of Fosha and Elgerd [6]. In employing the linear quadratic regulator theory, they introduced a significantly non-conventional approach which subsequently stimulated both an interest in the dynamic aspects of LFC as well as a lively and constructive dialogue between various control theorists and industry practitioners. One difficulty in this and other early studies is that they emphasized controlling synchronizing oscillations, whereas it is generally recognized that they are too fast for supplementary control to be either effective or desired [4]. Another weakness was recognized by Calovic [7], who made a clear distinction between the transient and steady state response aspects of LFC. A basic operating policy of interconnected control areas is that each area should attempt to regulate ACE to zero. While the relative gains applied to net interchange (M - S) and frequency deviation by area load-frequency controllers could perhaps be adjusted independently in order to obtain a desired transient response during normal operating conditions, the tie-line bias control strategy sets this relative gain equal to the frequency bias coefficient (B) for each area. This strategy thus achieves an implicit coordination between areas, which is very important during an abnormal operating condition, by essentially defining mutually determined and consistent steady state targets for all areas in the interconnection.

In the paper [8] the problem of distinguishing dynamic and steady state behavior was more fully explored and in addition a third major difficulty of earlier optimal regulator designs for LFC was addressed, namely the often dominating effect of prime mover energy source dynamics on unit generation response capability. Load-frequency control is fundamentally a tracking problem, and in the frequency range of interest the response limitations imposed by thermal energy sources must be recognized in the LFC design.

Based on these considerations, four basic objectives of LFC will be used to define the control problem:

1. The total area generation should be controlled so as to track the area load plus schedule. This will be called the load tracking objective.

2. The individual unit generations should be controlled so as to track the desired unit economic trajectories. This will be called the economic tracking objective.

3. Load-frequency control should allow the area primary response to occur naturally.

4. Individual unit response rate limitations must not be violated.

The parallel structure of LFC is exploited by designing independent controllers for each generating unit and a coordinating controller for the control area. These two components of the LFC structure are separately discussed in the next two sections.

3.0 COORDINATING CONTROLLER

The purpose of the coordinating controller is to provide megawatt reference inputs to the unit control loops such that the load and economic tracking objectives are achieved in a coordinated manner. Two basic design structures have been developed for the coordinating controller which correspond to two alternative methodologies for the design of robust linear multivariable regulators which have disturbance rejection properties. The first, which will be developed here in more detail, corresponds to the approach based on the feedback of estimates of (possibly artificial) disturbance states [9]. The second is based on the robust servo-compensator design methodology [10].

3.1 Model

The model used for developing the coordinating controller structure is a very simple representation of a control area connected to an external system. Tie-line synchronizing oscillations have been removed because, for the purposes of AGC simulation and design, it is reasonable to assume that the interconnection is at a common system frequency [11]. Table 1 contains definitions of the variables and parameters employed, while a block diagram is given in Figure 1. The closed loop system consisting of the unit and its unit controller is represented by a unity transfer function and thus the model is useful in the very low frequency range only.

3.2 Control Problem

With the white noise inputs v_1, v_2, v_3 and v_4 set to zero in Figure 1 it is necessary to determine state and control trajectories $\bar{x}$ and $\bar{u}$ which characterize the ultimate steady state. We require that the trajectories satisfy the equations corresponding to the model of Figure 1 and in addition that

$$ACE = y_2 + By_1 = 0 \tag{2}$$

These trajectories can be obtained as linear functions of the disturbances ω_1, ω_2, ω_3 and ω_4 and in fact many solutions are possible. Let γ be a set of participation factors such that $\gamma_1 + \gamma_2 + \dots + \gamma_n = 1$, let x^* be the desired economic unit trajectories determined by the dynamic dispatch functions, and denote the sum of the unit ultimate state trajectories by $\bar{x}_T$, i.e., $\bar{x}_T = \bar{x}_1 + \bar{x}_2 + \dots + \bar{x}_n$. Then the ultimate state and control trajectories are given by (3) and (4):

$$\bar{x} = x^* + \gamma\bar{x}_T \tag{3}$$

$$u = \gamma\bar{\omega}_4 \tag{4}$$

where:

$$\bar{x}_T = \frac{\beta}{B+\beta_2} S^o - \phi_1\omega_1 - \phi_2\omega_2 + \omega_3 \tag{5}$$

$$\phi_1 = \frac{B\beta}{B+\beta_2}, \quad \phi_2 = \frac{\beta_1 - B}{B+\beta_2} \tag{6}$$

A state variable gain matrix is obtained by next solving the deterministic optimal regulator problem corresponding to the pertubational system $\Delta x = x - x$, $\Delta u = u - \bar{u}$. Consider the sum $\Delta x_T = \Delta x_2 + \Delta x_2 + \dots + \Delta x_n$. Regulation of Δx_T during the transient actually implies regulation of total generation to its ultimate state value as established by the specification that ACE = 0.

Thus the scalar quadratic cost term $q_1(\Delta x_T)^2$ is a measure of performance for the load tracking objective of LFC defined previously. Furthermore, a quadratic weighting on variations of the quantities $\Delta x_1, \ldots, \Delta x_n$ in the subspace defined by $\Delta x_T = 0$ corresponds to a measure of performance for the economic tracking objective. Thus the state deviation costs in the performance index are separated into two terms which correspond to these two LFC tracking objectives:

$$J = \int_0^\infty [q_1 \Delta x^T C^T C \Delta \chi + \Delta \chi^T (I-C^*C)^T Q_2 (I-C^*C) \Delta x + \Delta u^T \bar{R} \Delta u] dt \qquad (7)$$

where

$$\bar{C} \varepsilon R^n, \ \bar{C}^T = [11 \ldots 1]$$

$$C \varepsilon R^{n+1}, \ C^T = [\bar{C}^T, 0]$$

and one choice for $C^* \varepsilon R^{n+1}$ is $(C^*)^T = [\bar{C}^*, 0]$, where $\bar{C}\,\bar{C}^* = I$. The (n+1) x(n+1) matrix Q_z is chosen to be block diagonal of the form. Q_2 = diagonal $(\bar{Q}_2, 0)$ where $\bar{Q}_2$ is n x n, in order to insure that the frequency mode, which is uncontrollable in a practical sense by a single area, is not observable in the cost functional.

The well-known solution to the standard regulator problem which provides the feedback gain matrix $\Delta u = -K \Delta x$ reduces to $K = [\bar{R}^{-1} P : 0]$ where $\bar{P}$ (n x n) is the solution of the relevant Ricatti equation. One interesting result of this solution is that the closed loop eigen-system can be shown to have n-1 modes which lie in the subspace corresponding to ACE=0 and an additional mode in the direction of $\bar{R}\bar{C}^T$. Moreover, it is this last mode which regulates ACE to zero and its response time is controlled by the scalar parameter q_1. The additional modes are "redistribution" modes which allow for the redistribution of generation to (possibly economic) targets which lie in the ACE=0 subspace and which have response times that depend on the parameter matrix $\bar{Q}_2$.

3.3 Simplified Regulator Design

The previous subsection described the characterization of ultimate state and control trajectories and the solution of a quadratic regulator problem which resulted in an n x n feedback gain matrix $R^{-1}\bar{P}$ that shapes the transient response of deviations about the ultimate state trajectories. Several practical problems remain, however. An important one is that the generating units on-line vary throughout the day and in addition, the system dispatcher and/or plant operators often change the operating mode of the on-line units. For the existing WE control center computer system it is considered infeasible to solve the Ricatti equation for the matrix $\bar{P}$ on-line when the set of units assigned to regulating duty changes, and there are in addition several practical problems associated with storing and retrieving n x n gain matrices for all possible combinations of regulating units.

The n-1 redistribution modes represent n-1 degrees of freedom for the regulator design in the sense that n-1 eigenvalues can be assigned to the secondary tracking objective (indirectly), via the parameter matrix $\bar{Q}_2$. Due to the parallel structure of the problem, it is possible to trade-off these n-1 degrees of freedom with the compuational burden of obtaining the feedback gain matrix K. While this interesting observation has not been fully explored in general, one particular case has been developed in detail. The special case in which the n-1 secondary tracking modes have a common eigenvalue corresponds to a solution of the Ricatti equation which can be obtained analytically. This allows gains to be updated very easily when the set of regulating units changes. For this special case the feedback gain matrix has the form

$$r_i = g_i \; g_i \; \ldots \; (g+g_i) \; g_i \ldots g_i \tag{8}$$

where r_i is the ith row of K and the term $(g+g_i)$ is in the ith column. Because off-diagonal terms are identical for each row, the resulting coordinating controller can be put into the very simple block diagram form of Figure 2.

3.4 Observor Design

The robustness and disturbance rejection properties of this coordinating controller arise from the feedback of the unmeasurable disturbance variables ω. The observor design used to provide estimates of these quantities is based on the methodology reported in [12]. Assigning a single time constraint ϕ^{-1} to the second order design, the resulting estimates are given in transfer function form below.

$$\hat{\omega}_1 = \left[\frac{1}{s\phi^{-1}+1}\right] \{[s(\phi^{-1}+H/\beta) +1] \; y_1 + 1/\beta_1 \; [(-D_1y_1+y_3)+y_2+S^o-y_4]\}$$

$$\hat{\omega}_2 = \beta_2/\beta \; [(-Dy_1+y_3) - y_4] + y_2 + S^o$$

$$\hat{\omega}_3 = -D_1y_1 + y_3$$

$$\hat{\omega}_4 = \frac{s}{s\phi^{-1}+1} \quad (-D_1y_1 + y_3) \tag{9}$$

3.5 Alternative Coordinating Controller Structure

An alternative to the coordinating controller structure discussed previously has been developed which is based on the robust servocompensator design approach [10]. Because these two design methodologies can in general lead to substantially different closed loop response characteristics, both are being developed with more or less equal emphasis in order to determine which structure is preferable for LFC.

A slightly different viewpoint is taken for this design which can be illustrated by defining the steady state relationships.

$$G_T \triangleq G_T^o - 1/R_T\Delta f, \quad L_T \triangleq L_T^o + \beta_L\Delta f \tag{10}$$

where G_T is the total area electrical generation, L_T is the total area electrical load at the prevailing frequency, G_T^o is the total area electrical generation at 60 Hz and L_T^o is the total area load demand. Now

$$ACE = \Delta IC + B\Delta f = G_T^o - L_T^o - S^o + (B - 1/R_T - \beta_L)\Delta f \tag{11}$$

and if for the sake of argument we assume that $B = 1/R_T + \beta_L$, then we see that ACE is a measure of the area electrical generation-load mismatch referenced to 60 Hz. Because the electric power which flows from a unit depends on the current electrical state of the entire system we can see from

(11) that the continuous regulation of ACE couples the various LFC area controllers through their objective functions. This can also be seen by expressing the net tie flow deviation in the form

$$\Delta IC = \frac{H_{EX}}{H_{SYS}} (P_T - L_T) - \frac{H_{AREA}}{H_{SYS}} (P_{EXT} - L_{EXT}) - S^o \qquad (12)$$

where P_T (P_{EXT}) is the total area (system) mechanical power and $H_{SYS} = H_{AREA} + H_{EX}$ is the total system inertia. Because of this argument, and the fact that mechanical power is actually controlled from the governor speed changer motor input, the load tracking objective of LFC is defined in terms of mechanical rather than electrical power for the alternative coordinating controller. The difference between the two is simply a derivative of frequency term, so the steady state strategy of ACE control is not affected, but this term in ACE is destabilizing and hence it is removed by defining

$$ACEM = P_T^o - L_T^o - S^o = ACE + 2H_{AREA}\, \dot{\Delta} f \qquad (13)$$

as the (mechanical) area control error which is to be regulated to zero. Whether or not this is a practical distinction is yet to be determined, but the relative magnitude of the derivative term and the performance of the filter designed (but not yet tested) to estimate mechanical power are the main factors.

In a manner similar to that described previously, the linear quadratic regulator design associated with the robust servocompensator design approach has been formulated, and again the special case which corresponds to an analytic solution of the relevant Ricatti equation has been obtained. A block diagram of the resulting coordinating controller is shown in Figure 3. It is interesting to note that, although designed using a modern control methodology, the resulting structure can be interpreted as a classical multi-loop integral controller with lead compensation.

4.0 UNIT CONTROLLER

The predominant dynamics of the load-frequency control problem reside in the low frequency domain associated with unit energy sources. Thus the response capability of a unit is a major factor in AGC performance and unit control is an important component of the LFC structure. The objectives of the unit controller are two-fold: regulate unit actual 60 Hz generation to track the unit desired generation as provided by the coordinating controller, and ensure that the unit's internal process variables are not unduly upset as a result of the tracking activity.

4.1 Control Problem

Reasonably detailed nonlinear models of the WE units have been developed for an AGC simulation program which will be used for further development and testing of the new AGC algorithms. A simplified and linearized model has been derived from these models for the purpose of design which is shown in Figure 4. It corresponds to a coal-fired drum unit whose controls are of the boiler-follower type.

An important observation concerns the interaction of the generation control loop, which is to be designed, and the pressure-fuel control loop. It can be shown that the transfer function from the speed-changer input to the generator power output may have a pair of non-minimum phase zeros for reasonable values of plant parameters. This would explain why increasing the generation control loop gain beyond some limit generally results in oscillatory behavior. It also would explain why it is practically impossible to improve performance by generation loop compensation methods alone. Thus, modern unit control systems employ coordination between these two strongly interacting control loops. Because throttle pressure is upstream of the possibly non-minimum phase zeros, one apparently effective solution to this problem is to use throttle pressure feedback.

The unit controllers have been designed using the linear quadratic gaussian method. Outputs of unit power output, throttle pressure deviation, frequency deviation and (for some units) first stage pressure are used by a Kalman filter for obtaining state variable estimates. The filter is designed to reference the estimate of unit output to 60 Hz in order to allow the unit's primary response to occur naturally. The controller contains a bias variable state in order to provide reset action and improve robustness. The feedback gain vector was obtained using standard optimal regulator techniques. In addition, various logic strategies involving hard constraints on selected variables and rates are also included.

The major difference between the LQG and conventional unit controller designs is in the incorporation of closed loop state variable estimates based on multiple measurements. The previous discussion emphasized the significance of measuring throttle pressure. Conventional designs often employ open loop estimation of a variable that represents boiler stored energy, which is a function of throttle pressure, and restrict control action when the estimated stored energy becomes too low. There is evidence that such an open loop estimate can be significantly in error, resulting in either an unnecessary restriction on unit response when the pressure is actually fine or unrealistic (and possibly destabilizing) control when the pressure is actually too low.

5.0 SUMMARY

The primary objectives of LFC have been briefly discussed in this paper, and a basic control structure derived in accordance with those objectives has been described. Some apparently reasonable assumptions have been used in order to derive, via methods of modern control theory, a control structure which is feasible for implementation on the existing WE control center computer system. Although similar in many respects to conventional designs, the resulting structure has several capabilities which, it is felt, can contribute to an improved AGC performance. A considerable amount of additional work is needed before a meaningful evaluation of the designs can be made, however.

Presently, the unit controller and the two coordinating controllers have been designed and partially evaluated using a combination of the LQG synthesis techniques, linear simulation and frequency domain analysis. Further development and evaluation of the LFC algorithms will in addition make extensive use of the detailed, non-linear AGC simulation program recently completed as a part of the project. In order to focus on LFC objectives and control structure, many design details and preliminary results have regretfully been omitted in this brief paper. Future reporting is intended however, that will discuss several interesting theoretical issues and important design details as well as provide results from the rather extensive evaluation which is planned.

6.0 ACKNOWLEDGEMENT

The advanced AGC development and implementation project is supported by the U.S. Department of Energy under contract number EC-77-01-2118. The direction and assistance provided by Lester H. Fink of DOE is gratefully acknowledged.

7.0 REFERENCES

1. N. Cohn, Control of Generation and Power Flow on Interconnected Systems, John Wiley and Sons, New York, 1966.

2. N. Cohn, "Some Aspects of Tie-Line Bias Control on Interconnected Power Systems", Trans. AIEE (Power Apparatus and Systems), Vol. 75, pp 1415-1428, February, 1957.

3. L.H. Fink, "Concerning Power System Control Structure", Advances in Instrumentation, ISA, Vol. 26, Part 1, 1971.

4. D.N. Ewart, "Automatic Generation Control - Performance Under Normal Conditions", Final Report for ERDA Contract E(49-18)-2147.

5. J.C. Russel and W.P. Rades, "The Wisconsin Electric Power Company Energy Control System", Proc. 1973 PICA Conference, Minneapolis, Minnesota, pp. 135-141, June 4-6, 1973.

6. C.E. Fosha and O.J. Elgerd, "The Megawatt - Frequency Control Problem: A New Approach Via Optimal Control Theory", IEEE Transactions, Vol. PAS-89, pp 563-578, April, 1970.

7. M.S. Calovic, "Linear Regulator Design for a Load and Frequency Control", IEEE Transactions, Vol. PAS-91, No. 6, pp 2271-2285, 1972.

8. H.G. Kwatny, K.G. Kalnitsky and A. Bhatt, "An Optimal Tracking Approach to Load-Frequency Control", IEEE Transaction, Vol. PAS-94, No. 5, pp 1635-1643, 1975.

9. H.G. Kwatny, "Optimal Linear Control Theory and a Class of PI Controllers for Process Control", Proceedings 13th JACC, pp 274-281, 1972.

10. E.J. Davidson, "The Output Control of Linear Time-Invariant Multivariable Systems with Unmeasurable Arbitrary Disturbances", IEEE Transactions, Vo. AC-17, pp 621-629, 1972.

11. C.W. Taylor and R.L. Cresap, "Real-Time Power System Simulation for Automatic Generation Control", presented at the IEEE PES Summer Meeting, San Francisco, CA, 1975.

12. H.G. Kwatny, "Minimal Order Observors and Certain Singular Problems of Optimal Estimation and Control", IEEE Transactions, Vol. AC-19, No. 3, pp 274-276, 1974.

TABLE 1
NOMENCLATURE

Δf, y_1	frequency deviation from nominal
ΔIC, y_2	net tie flow deviation from schedule
y_3	area electrical load (at prevailing frequency)
P_T^o , y_4	area mechanical power at 60 Hz
H	total system inertia
β_1	local area regulation $\beta_1 = D_1 + R_1^{-1}$
β_2	external area regulation $_2 = (D_2 + R_1^{-1})$
β	total system regulation $(\beta_2 = \beta_1 + \beta_2)$
D_1	local area load characteristic
D_2	external area load characteristic
R_1	local area governor characteristic
R_2	external area governor characteristic
B	frequency bias (positive) constant used in definition of ACE
ACE	area control error, ACE $\triangleq \Delta IC + B\Delta f$
x_i	ith unit generation demand @ 60 Hz
x_{n+1}	state variable representing frequency under ideal conditions
ω_1	disturbance term representing frequency error due to imperfect knowledge of β
ω_2	disturbance term representing external area generation and load conditions
ω_3	disturbance representing local area electrical load at 60 Hz
ω_4	disturbance representing rate of change of local area load
u_i	ith unit control variable, rate of change of generation
Δ_1	local area excess generation at 60 Hz
Δ_2	external area excess generation at 60 Hz
S^o	local area net tie flow schedule

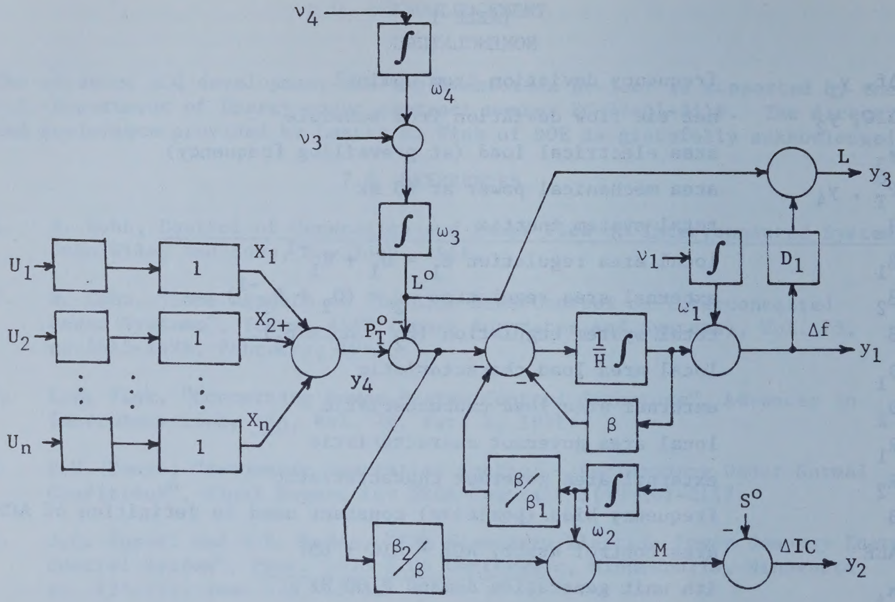

ν4
ω4
ν3
L
y3
ω3
U1
1
X1
ν1
D1
L°
ω1
U2
1
X2+
P_T^o -
1/H ∫
Δf
y1
y4
β
Un
1
Xn
β/β1
S°
ω2
β2/β
M
ΔIC
y2

FIGURE 1: MODEL FOR COORDINATING CONTROLLER DESIGN

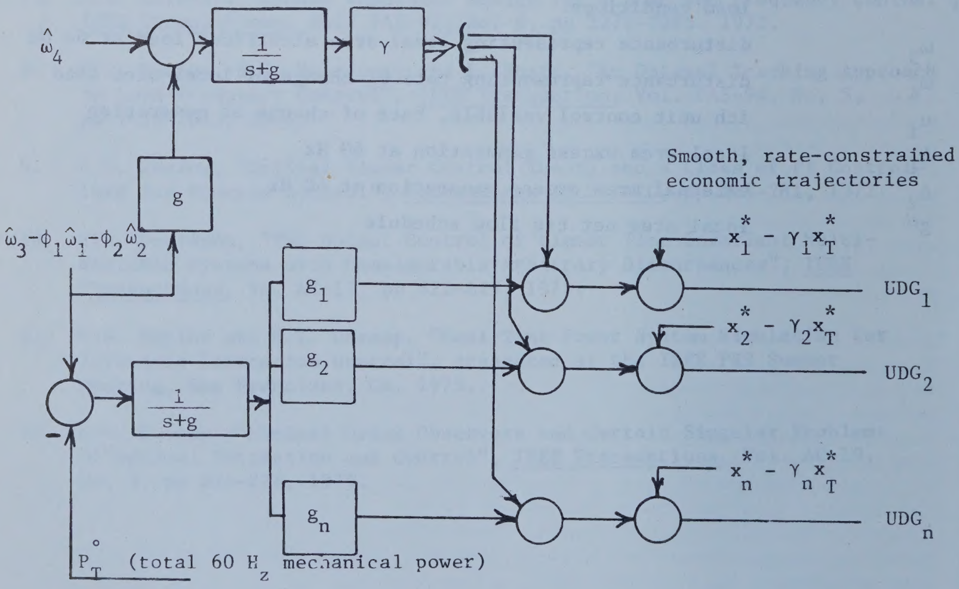

$\hat{\omega}_4$
1/(s+g)
γ
g
Smooth, rate-constrained economic trajectories
$\hat{\omega}_3-\phi_1\hat{\omega}_1-\phi_2\hat{\omega}_2$
$x_1^* - \gamma_1 x_T^*$
g1
UDG1
$x_2^* - \gamma_2 x_T^*$
g2
UDG2
1/(s+g)
-
$x_n^* - \gamma_n x_T^*$
gn
UDGn
P_T° (total 60 Hz mechanical power)

FIGURE 2: SIMPLIFIED COORDINATING CONTROLLER

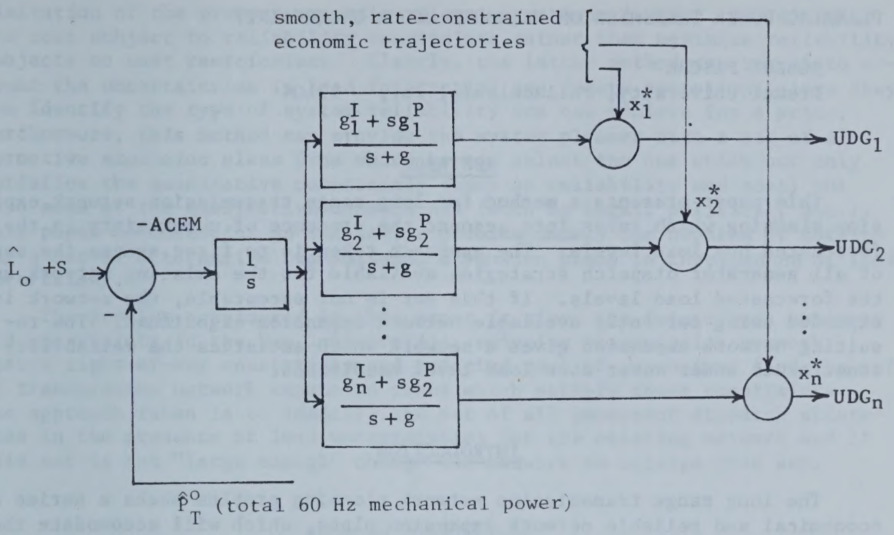

smooth, rate-constrained economic trajectories
x_1^*
x_2^*
x_n^*
$\frac{g_1^I + sg_1^P}{s+g}$
$\frac{g_2^I + sg_2^P}{s+g}$
$\frac{g_n^I + sg_2^P}{s+g}$
UDG$_1$
UDC$_2$
UDG$_n$
-ACEM
$\hat{L}_o$ +S
$\frac{1}{s}$
−
$\hat{P}_T^o$ (total 60 Hz mechanical power)

FIGURE 3: ALTERNATIVE COORDINATING CONTROLLER

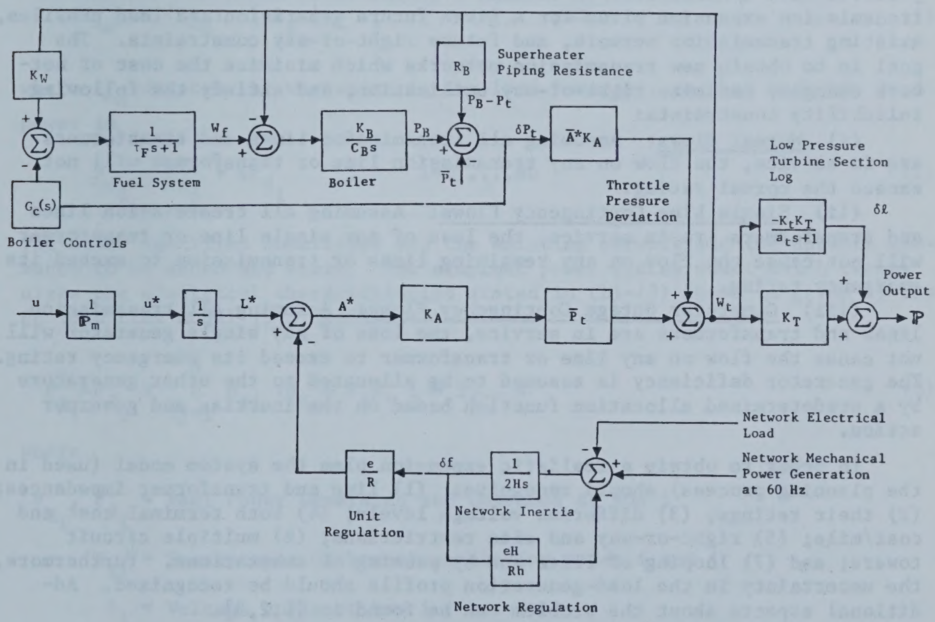

K$_W$
+
−
Σ
$G_c(s)$
Boiler Controls
$\frac{1}{\tau_F s+1}$
Fuel System
W$_f$
$\frac{K_B}{C_B s}$
Boiler
P$_B$
R$_B$
Superheater Piping Resistance
P$_B$ − P$_t$
$\bar{P}_t$
δP$_t$
$\bar{A}$*K$_A$
Throttle Pressure Deviation
Low Pressure Turbine Section Log
$\frac{-K_t K_T}{a_t s+1}$
δℓ
Power Output
u
$\frac{1}{P_m}$
u*
$\frac{1}{s}$
L*
A*
K$_A$
$\bar{P}_t$
W$_t$
K$_T$
P
Network Electrical Load
Network Mechanical Power Generation at 60 Hz
$\frac{1}{2Hs}$
Network Inertia
δf
$\frac{e}{R}$
Unit Regulation
$\frac{eH}{Rh}$
Network Regulation

FIGURE 4: MODEL FOR UNIT CONTROLLER DESIGN

PLANNING POWER TRANSMISSION NETWORKS UNDER UNCERTAINTY

ROBERT FISCHL
Drexel University, Philadelphia, Pa., 19104

ABSTRACT

This paper presents a method for long range transmission network expansion planning which takes into account the presence of uncertainty in the forecasted bus load levels. The approach taken is to first assess the set of all generator dispatch strategies available for the existing network using the forecasted load levels. If this set is not acceptable, the network is expanded using currently available network expansion algorithms. The resulting network expansion gives a network which satisfies the reliability constraints under worst case load level conditions.

INTRODUCTION

The long range transmission network planning problem seeks a series of economical and reliable network expansion plans, which will accomodate the future generation and load growth, over the planning period T. The basic objective is to find the type, location, and timing of the network additions (or changes) at minimum cost subject to reliability environmental, legal, and political constraints. Since many factors are difficult to quantify and predict, this problem is extremely difficult to formulate mathematically.

Currently, the transmission network expansion optimization problem objectives have concentrated on seeking a series of economical and reliable transmission expansion plans for a given future generation and load profiles, existing transmission network, and future right-of-way constraints. The goal is to obtain new transmission networks which minimize the cost of network changes, maximize right-of-way utilization, and satisfy the following reliability constraints:

(i) Normal Flows: Assuming all transmission lines and transformers are in service, the flow on any transmission line or transformer will not exceed the normal rating.

(ii) Single Line Contingency Flows: Assuming all transmission lines and transformers are in service, the loss of any single line or transformer will not cause the flow on any remaining lines or transmission to exceed its emergency rating.

(iii) Generator Outage Contingency Flows: Assuming all transmission lines and transformers are in service, the loss of any single generator will not cause the flow on any line or transformer to exceed its emergency rating. The generator deficiency is assumed to be allocated to the other generators by a predetermined allocation function based on the inertias and governor action.

In order to obtain a realistic expansion plan the system model (used in the planning process) should recognize: (1) line and transformer impedances; (2) their ratings; (3) different voltage levels; (4) both terminal cost and cost/mile; (5) right-or-way and site restrictions; (6) multiple circuit towers; and (7) looping of lines and by passing of substations. Furthermore, the uncertainty in the load-generation profile should be recognized. Additional aspects about the problem can be found in [1,2,3].

There are several planning algorithms available [4-9] for the solution of the above problem, each based on a different interpretation of the system model and the choice of the design objective. Probably the most important limitation of the present expansion planning methods is that they minimize the cost subject to reliability constraints rather than maximize reliability subjects to cost restrictions. Clearly, the latter method can take into account the uncertainties in load forecasting and future operation, since they can identify the type of system reliability one can achieve for a price. Furthermore, this method can provide the system planner with a set of alternative expansion plans from which he can select the one which not only satisfies the quanitative constraints (such as reliability and cost) but also some of the qualitative constraints (such as legal, political, etc.). This paper concentrates on the latter problem, namely the problem of obtaining a set of alternative transmission expansion plans in thepresence of load uncertainty.

The problem considered in this paper is given the future best estimate and uncertainty in the bus load profile, existing transmission network, future right-of-way constraints and budgeted cost of expansion; find the set of transmission network expansion plans which satisfy these constraints. The approach taken is to identify the set of all generator dispatch strategies in the presence of load uncertainties for the existing network and if this set is not "large enough" change the network to enlarge this set.

PROBLEM FORMULATION

The approach taken to the solving the transmission network planning in the presence of load uncertainties is depicted in Fig. 1. One of the specified quantities are the forcasted bus load profile, $\underline{P}_d$, in terms of its best-estimated value, $\underline{P}^\circ_d$, and the range of variation denoted by $U(\delta\underline{P}_d)$ and given by

$$U(\delta\underline{P}_d) = \{\delta\underline{P}_d : \ |\delta P_{d_i}| \leq \varepsilon_i \quad ; \quad i=1,...,ND\} \tag{1}$$

where $\delta\underline{P}_d$ denotes the variation in bus load forecast, so that each bus load power is

$$\underline{P}_{d_i} = P^\circ_{d_i} + \delta P_{d_i} \quad , \qquad i=1,...,ND \tag{2}$$

The other specified quantities are the existing network and the network elements to be added and where. The simplest power system model which recognizes the electrical characteristics stated in (1)-(3) above is given by the dc load flow model [5], namely.

$$P_{g_i} - P_{d_i} = \sum_{j=1}^{N} B_{ij} \ (\delta_i - \delta_j), \ i = 1,...N \tag{3}$$

where

P_{g_i}, P_{d_i} = Real Power Generated and Consumed at Bus i

B_{ij} = Susceptance of branch connecting nodes i and j

δ_i = Voltage angle at bus i

N = Total number of buses

In the steady-state, the reliability constraints represent the thermal

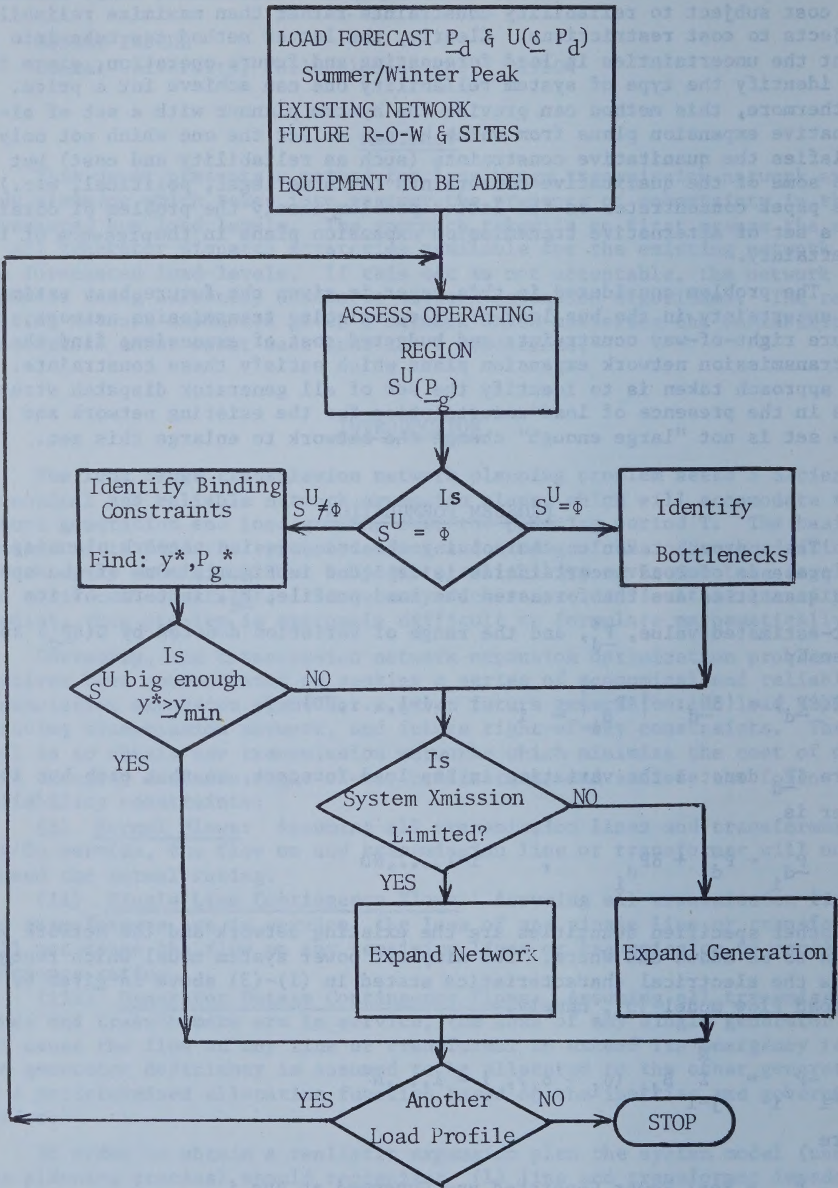

LOAD FORECAST P_d & $U(\delta P_d)$
Summer/Winter Peak
EXISTING NETWORK
FUTURE R-O-W & SITES
EQUIPMENT TO BE ADDED
ASSESS OPERATING REGION $S^U(P_g)$
Is $S^U = \Phi$
$S^U \neq \Phi$
$S^U = \Phi$
Identify Binding Constraints
Find: y^*, P_g^*
Identify Bottlenecks
Is S^U big enough $y^* \geq y_{min}$
NO
YES
Is System Xmission Limited?
NO
YES
Expand Network
Expand Generation
Another Load Profile
YES
NO
STOP

Figure 1: Transmission Network Planning Algorithm.

limits on the transmission element which can be stated mathematically by the inequality constraints:

$$\begin{aligned} |P_\mu| &\le R^N_\mu && \text{for all } \mu \\ |P_{\mu m}| &\le R^E_\mu && \text{for all } \mu \text{ and } m \end{aligned} \tag{4}$$

where P_μ is real power flow in branch μ under normal operation; $P_{\mu m}$ is the real power flow in branch μ due to single contingency outage of line m (or generator m); and R^N_μ, R^E_μ are the normal and emergency thermal ratings of branch μ, respectively. Note that the branch flows P_μ and $P_{\mu m}$ are functions of the network susceptances and generation-load profile given in (3) which can be represented by

$$\underline{P}_\mu = [H(\underline{B})]\ (\underline{P}_g - \underline{P}_d) \tag{5}$$

where [H] is an LxN transfer matrix relating branch flows to net bus injections, and L is the total number of network branches.

The first step in the algorithm shown in Fig. 1 is to determine if there exists a set of secure generations $\underline{P}_g$, denoted by $S^U(P_g)$ for which none of the constraints are violated for all load variations $\delta\underline{P}_d$ in U defined in (1). The definition of the security region S^U is given in [10] and is summarized here. First, consider the transmission constraints of (4). For each choice of variations $\underline{\delta P}_d \varepsilon U$, the perturbed normal transmission constraint set, denoted by N^T, is given by:

$$N^T(\delta\underline{P}_d) = \{\underline{P}_g:\ -\underline{R}^N + H_d\ (\underline{P}^\circ_d + \delta\underline{P}_d) \le H_g\ \underline{P}_g \le \underline{R}^N + H_d\ (\underline{P}^\circ_d + \delta\underline{P}_d)\} \tag{6}$$

where $[H_d]$ and $[H_g]$ represent submatrices of [H]. In this paper it will be assumed that the load levels are uniformly distributed about their mean values. Hence, each choice of $\delta\underline{P}_d$ is equally likely and all possible sets $N^T(\delta\underline{P}_d)$ must be considered for all $\delta\underline{P}_d \varepsilon U$. In general, the uncertainties in the load profile could follow other statistical distributions and thus would require a more complex analysis than that presented here. In the absence of further statistical information, the choice of uniform distribution is assumed to be adequate. Under the stated assumptions, we can define a permanently secure region for the normal set of transmission constraints in the presence of uncertainty in the load profile. This permanently secure region region, denoted by N^{TU}, represents the intersection of all sets $N^T(\delta\underline{P}_d)$ over all $\delta\underline{P}_d \varepsilon U$, namely,

$$N^{TU} = \bigcap_{\delta\underline{P}_d \varepsilon U} N^T(\delta\underline{P}_d) \tag{7}$$

If the load variations δP_{di} are assumed to be independent, then the permanently secure normal set of transmission constraints, denoted by N^{TU}, is given by:

$$N^{TU} = \{\underline{P}_g:\ -\underline{R}^N + H_d\ \underline{P}^\circ_d + \underline{\pi} \le H_g\ \underline{P}_g \le \underline{R}^N + H_d\ \underline{P}^\circ_d - \underline{\pi}\} \tag{8}$$

where $\underline{\pi}$ is a non-negative vector whose components are given by

$$\pi_i = \sum_{j=1}^{N_d} |h_{ij}\varepsilon_j| \tag{9}$$

h_{ij} = (i,j)-th element of matrix H_d

ε_j = maximum deviation from the mean of the load level at load node j.

Similarly, permanently secure region can be defined for all the single contingency outages [10].

In addition to the transmission constraints, there is a set of constraints imposed by the generator operating characteristics and the available reserves that can be assigned under postulated generator outage contingencies. This set is not affected by the load profile and is therefore independent of any uncertainty in the load levels. The two sets of constraints are defined in [10] by N^G and C^G:

$$N^G = \{\underline{P}_g : \underline{P}_g^{min} \leq \underline{P}_g \leq \underline{P}_g^{max}\} \tag{10}$$

$$C^G = \{\underline{P}_g : \sum_{\substack{i=1 \\ i\neq k}}^{N} \text{Min}\,\{\alpha_i T_i P_{gi}^{max},\ P_{gi}^{max} - P_{gi}\} \geq G_k,\ k=1,\ldots M\} \tag{11}$$

The permanently secure regions, S^U, which identify the normal secure, normal insecure, and emergency operating states, characterized by DyLiacco [11], in the presence of the uncertainty in the load profile can now be defined by combining the appropriate sets from those defined above. Thus the normal secure region under uncertainty in the load, is given by:

$$S_{NS}^U = N^{TU} \cap N^G \cap C^{TU} \cap C^G \cap C^{GU} \tag{12}$$

where C^{TU} and C^{GU} denote the permanently secure set of constraints due to postulated transmission elements and generator contingency outages, respectively.

It should be noted that the permanently secure regions under uncertainty, are contained in the security regions assuming no uncertainty. For example,

$$S_{NS}^U \subseteq S_{NS}(0) \tag{13}$$

where $S_{NS}(0)$ denotes the case when $\underline{\delta P_d} = \underline{0}$, namely, the case when the load levels are known exactly. This is the result of the definition of the permanently secure region of (7). Furthermore, note that $S_{NS}^U \subseteq S_{NS}(\delta \underline{P_d})$ for all $\delta P_d \varepsilon U$ and since each set $S_{NS}(\delta \underline{P_d})$ is convex and polyhedral then S_{NS}^U is also convex and polyhedral.

The set of constraints which define any one of the permanently secure regions defined above, can be written in the general form,

$$S^U = \{\underline{x}:\ [A]\underline{x} \leq \underline{b}^\circ - \underline{\beta}\} \tag{14}$$

where [A] is an mxn matrix of real coefficients; $\underline{x}$ is an nxl vector of real operating variables; $\underline{b}^\circ$ is an mxl vector of real constants; and $\underline{\beta}$ is an mxl vector defining the range of variation of $\underline{b}$, i.e., $-\beta_i \leq \delta b_i \leq \beta_i$

As indicated in Fig. 1 there are a number of questions which must be resolved before adding transmission elements to the network. The first question is concerned with the existence of a nonempty region defined by (14). If the region S^U is empty then the network must be changed to obtain a set of secure operating points. If the region S^U is not empty then the question is concerned with the "size" of the region. As discussed in [10] one can define a measure of the size of S^U by

$$y(\underline{x}) = \min_{1 \leq i \leq m} \{d_i(\underline{x})\} \tag{15}$$

where $d_i(\underline{x})$ denotes the orthogonal deviation of the point $\underline{x}$ from the hyperplane defined by the i-th constraint, namely

$$d_i(\underline{x}) = \frac{1}{t_i} (b_i - \sum_{j=1}^{n} a_{ij} x_j)$$

where

$$t_i = (\sum_{i=1}^{n} a_{ij}^2)^{\frac{1}{2}} \quad ; \quad i=1,2,\ldots,m \tag{16}$$

Note that $d_i(x) \geq 0$ for $\underline{x}$ in S, and when $\underline{x}$ is not in S then some of the $d_i(\underline{x}) \leq 0$.

Using the measure $y(\underline{x})$ of (15) one can find the "size" of S^U by find the maximun value of y for all $\underline{x}$ in S, namely

$$y^* = y(\underline{x}^*) = \max_{x \in S^U} \{y(\underline{x})\} \tag{17}$$

Since the constraints defining S^U are linear, then y^* can be obtained using standard linear programming (LP) methods. The solution of this LP problem gives an $\underline{x}^*$ or a generation dispatch point and also indentifies the binding constraints. If all the binding constraints are in the set $N^G \cap C^G$, i.e. constraints imposed by the generator operating characteristics then changes in the network will not affect the region S^U. Therefore, as shown in Fig. 1, the transmission network is expanded only if the generator operating region, S^U, is transmission limited.

PROBLEM SOLUTION

The operating region assessment problem depicted in Fig. 1, is solved by the algorithm shown in Fig. 2. It involves the solution of one of the following LP problems, depending on whether S^U is empty or non-empty:

Problem A1: (S non-empty)

Maximize y subject to the constraints,

$$A\underline{x} + \underline{t}y \leq \underline{b}$$
$$\underline{x} \geq \underline{0} \quad ; \quad y \geq 0 \tag{18}$$

where t_i is given by Eq. (16) and $\underline{b} = \underline{b}^\circ - \underline{\beta}$ if the permanently secure region S^U is to be identified and $\underline{b} = \underline{b}^\circ$ if the security region S based on the best estimate is being sought. Note that this LP problem is defined in (n+1) space and the constraints define an (n+1)-polytope in E^{n+1}. Fig. 3 illustrates the region S of a 6-node and 8 branch test system having 2 generators [12]. Out of the 16 inequality constraints only 5 are binding. These are identified by post-processing in the Simplex Tableau as shown in [12].

Problem A2: (S empty)

Minimize $\sum_{i=1}^{m} v_i$ subject to the constraints,

$$\sum_{j=1}^{n} a_{ij} x_j - t_i v_i \leq b_i \quad , \quad i=1,\ldots,m$$
$$\underline{x} \geq 0 \quad , \quad \underline{v} \geq 0$$

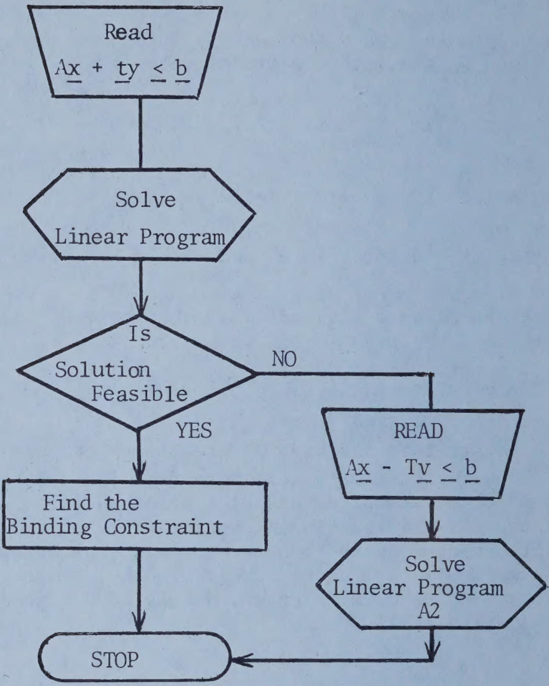

Read
Ax + ty ≤ b
Solve
Linear Program
Is
Solution
Feasible
NO
YES
READ
Ax - Tv ≤ b
Find the
Binding Constraint
Solve
Linear Program
A2
STOP

Figure 2: Flow Chart of Identification Algorithm.

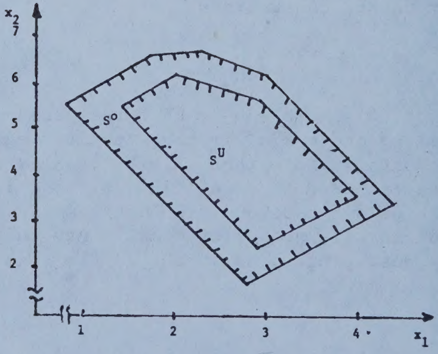

x2
7
6
5
4
3
2
S°
S^U
1
2
3
4
x1

Figure 3: Two Dimensional Representation of the Operating Region, S.

where $v_i = -d_i$, defined in (16). Recall that if $S = \Phi$ then there exist no x which satisfies the constraint set (14), however by introducing the variables v_i, one guarantees the solution of the LP problem A2.

The solution of either one of the LP problems defined above gives the operating point $\underline{x}^*$, that is a generation dispatch strategy P_g^*, which can be used in applying any of the standard network expansion methods [6,13,14]. There is no problem in directly applying these methods when the operating region S^U is empty. If the region S^U then the operating point P_g^* guarantees that there are no network elements which are overloads so that the network expansion methods must be modified. Recall from Fig. 1 that for this case the criterion for having an acceptable network is that the measure, y^*, of the size of the region S^U, is greater than some minimum value y^*_{min} namely the condition

$$y^* \geq y^*_{min} \tag{20}$$

If this condition is not satisfied, then the reliability constraints of (4) which are used in the network expansion problem should be modified as follows:

$$|P_\mu \; (\underline{P}_g^*, \underline{P}_d^\circ)| \leq R_\mu^N - (\pi_\mu + t_\mu \; y^*_{min}) \quad , \quad \text{for all } \mu \tag{21}$$

where P_μ represents the line flow in branch μ when $P_g = P_g^*$ and $P_d = P_d^\circ$ in (5). The term $(\pi_\mu + t_\mu \; y^*)$ reduces the ratings of the network elements in order to guarantee that some elements in the network are overloaded.

Once the generation and load profile has been specified, then the optimization method of [13] is applied to identify the set of feasible networks which satisfy all the constraints dictated by the thermal constraints of (4) or (21), the right-of-way restrictions, and the budget restrictions. Any one of these networks is guarnateed to satisfy the constraints for all load levels given by (2) when the uncertainties are within the set U.

CONCLUSIONS

This paper has presented a method for tranmission network expansion planning in the presence of uncertainties in the forecasted load profile. The concept of first assessing the set of all secure dispatch strategies under load uncertainties allows one to solve the network expansion using the currently available deterministic optimization methods. Furthermore, the operating region allows one to prefilter the number of constraints one must consider in using the penalty function approach to the solution of the constrained optimization problem describing the transmission network expansion problem.

Although the present work in incorporating the uncertainties into the planning process has concentrated on essentially the worst-case conditions (by the nature of the load uncertainty model), future work will consider a probalistic load model, in which the bus loads are described by their statistics. This requires the reformulation of the secure operating region in terms of probabilistic (or chance) constraints, and then find their deterministic equivalents for a prespecified level of reliability.

ACKNOWLEDGEMENT

This work was funded in part by the Division of Electric Energy Systems, U.S. Department of Energy, to whom grateful acknowledgement is made.

REFERENCES

[1] Knight, U.G., Power System Engineering and Mathematics, Pergamon, New York, 1972.

[2] Knight, U.G., R.R. Booth, S.A. Mallard and D.M. Lewis, "Computers in Power System Planning", Proc. of IEEE, Vol. 62, No. 7, pp 872-882, July 1974.

[3] Fischl R., "Optimal System Expansion: A Critical Review", Systems Engineering for Power: Status and Prospects, Proc. ERDA Conf-750867, pp. 233-260, August 1975.

[4] Garver, L.L., "Transmission Network Expansion Using Linear Programming", IEEE Trans. Power App. and Syst. Vol. PAS-89, pp 1688-1697, September/October 1970.

[5] Fischl, R. and W.R. Puntel, "Computer-Aided Design of Electric Power Transmission Networks", Paper No. C72-168-8, IEEE Winter Meeting, New York, N.Y., January 1972.

[6] Puntel, W.R., N.D. Reppen, R.J. Ringlee, J.E. Platts, W.A. Ryan and P.J. Sullivan, "An Automated Method for Long-Range Planning of Transmission Networks", PICA Conference Procedings: 38-46, June 1973.

[7] Lee, S.Y., K.L. Hides and E. Hnyilicza, "Transmission Expansion by Branch-and-Bound Integer Programming with Optimal Cost-Capacity Curves", IEEE Trans. Power App. and Syst., Vol. PAS-93, pp 1390-1400, September/October 1974.

[8] Adams, R.N. and M.A. Laughton, "Optimal Planning of Power Networks Using Mixed-Integer Programming Part I-Statric and Time-Phased Network Synthesis", Proc. IEEE, Vol. 121, No. 2, pp 139-147, February 1974.

[9] Dusondhet, Y.P. and A. El-Abiad, "Transmission Planning Using Discrete Dynamic Optimizing", IEEE Trans. Power App. and Syst., Vol. PAS-92, No. 4, pp 1358-1371, July/August 1973.

[10] Fischl, R., G.C. Ejebe and J.A. DeMaio, "Identification of Power System Steady-State Security Regions Under Load Uncertainty", Paper No. A 76-495-2, IEEE PES Summer Meeting, Portland, 1976.

[11] DyLiacco, T.E., "Real-Time Computer Control of Power Systems", Proc. IEEE, Vol. 62, pp. 884-891, 1974.

[12] DeMaio, J.A. and R. Fischl, "Fast Identification of the Steady-State Security Regions for Power System Security Enhancement", Paper No. A-76-076-0, IEEE Winter Power Meeting, January 1976.

[13] Fischl, R. and Ho, T.F., "Identification of a Set of Reliable Transmission Network Expansion Plans Within a Given Budget: Parts I and II, 1977 PICA Meeting, Toronto, Canada.

[14] Fischl, R., Huckins, W.J., et al. "Methods in Transmission Network Expansion Planning", Paper No. A-77-504-4 IEEE PES Summer Meeting, Mexico, 1977.

MODEL REDUCTION OF DYNAMIC SYSTEMS OVER A FREQUENCY INTERVAL

G. LANGHOLZ*
Department of Electrical Engineering
and Computer Science
University of California
Santa Barbara, Ca 93106

Y. BISTRITZ
School of Engineering
Tel-Aviv University
Tel-Aviv
Israel

ABSTRACT

Model reduction of linear, time-invariant SISO systems over desired frequency intervals (low-pass, band-pass and high-pass) is considered in this paper. Order reduction is effected by manipulating two orthogonal polynomial series, one representing the high-order system and the other representing the approximating low-order model. The technique enables the selection of stable and minimal phase reduced-order models.

INTRODUCTION

Order reduction of dynamic systems is an area of research which receives considerable attention in the literature. It is an important subject from both theory and engineering applications viewpoints [1].

Of the various model reduction techniques, the Padé class of methods, e.g. [2-4], are computationally more attractive. However, two major problems are associated with these methods. First, if the dynamic system is stable, its reduced order model might be unstable. Second, most approximations are about a single frequency point ($s \to 0$) which may yield poor whole frequency response. Methods to solve these problems were suggested in [4-7] for example.

In this paper, reduced-order modeling of linear, time-invariant systems characterized by transfer function is accomplished over a desired frequency interval, low-pass, band-pass or high-pass. The method is based on manipulating two orthogonal polynomial series, modified Chebyshev polynomials, one representing the high-order system and the other representing the approximating low-order model.

The proposed method can be viewed as a generalization of the classical Padé approximations. However, by a special transformation it becomes the (simple) classical Padé problem. Thus, the method combines the superiority of the Chebyshev polynomial expansion over a frequency interval with the simplicity and flexibility of the classical Padé approximation. In addition, the technique enables the selection of stable and minimal phase reduced-order models.

MATHEMATICAL PRELIMINARIES

Darlington Polynomials

Consider the following n-th order modified Chebyshev polynomials, referred to in the sequel as Darlington polynomials [8] - see Appendix

$$D_n(x) = \cos(n\sin^{-1}x) \quad , \quad n \text{ even} \qquad (1a)$$

$$= j\sin(n\sin^{-1}x) \quad , \quad n \text{ odd} \qquad (1b)$$

which are orthogonal on $x \in [-1,1]$ with weight function $\left(1-x^2\right)^{-\frac{1}{2}}$.

Let $[0,\omega_0]$ be a given frequency interval and substitute ω/ω_0 for x in (1). Hence

$$D_n(\omega/\omega_0) = \cos(n\phi) \quad , \quad n \text{ even} \qquad (2a)$$

$$= j\sin(n\phi) \quad , \quad n \text{ odd} \qquad (2b)$$

where $\phi = \sin^{-1}(\omega/\omega_0)$.

---

* On sabbatical leave from School of Engineering, Tel-Aviv University, Tel-Aviv, Israel.

The ζ-plane

Define a new variable $\zeta = \exp(j\phi)$, then, with $s=j\omega$, the relationship between the s-plane and the (new) ζ-plane is

$$s = \omega_0(\zeta - 1/\zeta)/2 \tag{3}$$

It can be easily shown that every point s_α in the s-plane maps into two points in the ζ-plane, ζ_α and $-1/\zeta_\alpha$, one being inside and the other being outside the unit circle. To obtain a one-to-one mapping, ζ in (3) will be restricted in the sequel, $|\zeta|>1$. Hence, (3) is defined for every $s \notin [-j\omega_0, j\omega_0]$, but, nevertheless, maintains $s_\alpha \leftrightarrow \zeta_\alpha \Longrightarrow \bar{s}_\alpha \leftrightarrow \bar{\zeta}_\alpha$ (the bar denotes complex conjugate) and $\operatorname{sgn}\left(\operatorname{Re}\{s_\alpha\}\right) = \operatorname{sgn}\left(\operatorname{Re}\{\zeta_\alpha\}\right)$.

Rewriting Darlington polynomials in ζ yields

$$D_n(\zeta) = \frac{1}{2}\left[\zeta^n + \left(\frac{1}{\zeta}\right)^n\right], \quad n \text{ even} \tag{4a}$$

$$= \frac{1}{2}\left[\zeta^n - \left(\frac{1}{\zeta}\right)^n\right], \quad n \text{ odd} \tag{4b}$$

or, combined together

$$D_n(\zeta) = \frac{1}{2}\left[\zeta^n + \left(-\frac{1}{\zeta}\right)^n\right], \forall n \tag{5}$$

System Representation In The ζ-plane.

Let {m/n} denote the rational function

$$G(s) = K\prod_{i=1}^{m}(s-s_i{}')\Big/\prod_{i=1}^{n}(s-s_i{}''), \quad m<n \tag{6}$$

that characterizes a linear, time-invariant dynamic system. Using (3), G(s) maps into $Q(\zeta) = R(\zeta)\,R(-1/\zeta)$ over the frequency interval $[0,\omega_0]$, where

$$R(\zeta) \triangleq K_\zeta^{\frac{1}{2}}\prod_{i=1}^{m}(\zeta-\zeta_i{}')\Big/\prod_{i=1}^{n}(\zeta-\zeta_i{}'') \tag{7}$$

$$K_\zeta \triangleq K\prod_{i=1}^{m}(-\omega_0/2\zeta_i{}')\Big/\prod_{i=1}^{n}(-\omega_0/2\zeta_i{}'') \tag{8}$$

Let g and ψ be the gain and phase respectively of G(s), $\ln G(s) = g+j\psi$. Thus, it can be shown that in the ζ-plane

$$g+j\psi = \ln R(\zeta) + \ln R(-1/\zeta) \tag{9}$$

the gain of G(s) being

$$g = \frac{1}{2}\ln R(\zeta)R(-\zeta) + \frac{1}{2}\ln R\left(-\frac{1}{\zeta}\right)R\left(\frac{1}{\zeta}\right) \tag{10}$$

and the phase of G(s) being

$$j\psi = \frac{1}{2}\ln R(\zeta)/R(-\zeta) + \frac{1}{2}\ln R\left(-\frac{1}{\zeta}\right)/R\left(\frac{1}{\zeta}\right) \tag{11}$$

Note that while (9) is valid for every s and ζ, (10) and (11) are only valid for $s=j\omega, |\zeta| = 1$.

Darlington Polynomial Expansion Of Transfer Functions In The ζ-Plane.

Given G(s) of (6), consider a Darlington polynomial series expansion of $g+j\psi$ over the frequency interval $[0,\omega_0]$

$$g+j\psi = \sum_{i=0}^{\infty} d_i D_i(\omega/\omega_0) \tag{12}$$

Mapping (12) into the ζ-plane, using (10) for its left-hand side and (5) for its right-hand side, yields

$$\ln R(\zeta) + \ln R(-1/\zeta) = \frac{1}{2}\sum_{i=0}^{\infty} d_i\zeta^i + \frac{1}{2}\sum_{i=0}^{\infty} d_i(-1/\zeta)^i \tag{13}$$

Thus, it can be shown [9] that the problem of Darlington polynomial series expansion of $\ln G(s)$ in the s-plane is transformed in the ζ-plane into a problem of power series expansion

$$\ln R(\zeta) = \frac{1}{2}\sum_{i=0}^{\infty} d_i \zeta^i \tag{14}$$

where it is noted that $R(\zeta)$ is structured analogously to $G(s)$. Similarly, a Darlington series expansion in the s-plane of the gain of $G(s)$

$$\frac{1}{2}\ln G(s)G(-s) = \sum_{i=0}^{\infty} d_{2i} D_{2i}(\omega/\omega_0) \tag{15}$$

is a power series in the ζ-plane

$$\ln R(\zeta)R(-\zeta) = \sum_{i=0}^{\infty} d_{2i}\zeta^{2i} \tag{16}$$

and a Darlington series expansion in the s-plane of the phase of $G(s)$

$$\frac{1}{2}\ln G(s)/G(-s) = \sum_{i=0}^{\infty} d_{2i+1} D_{2i+1}(\omega/\omega_0) \tag{17}$$

is a power series in the ζ-plane

$$\ln R(\zeta)/R(-\zeta) = \sum_{i=0}^{\infty} d_{2i+1}\zeta^{2i+1} \tag{18}$$

Note that while (12) is valid for every s and ζ, (14), (16) and (18) are only valid for $|\zeta| = 1$ and (15) and (17) are only valid for $s=j\omega$.

An algorithm for deriving the d_i coefficients is outlined in [9].

MODEL REDUCTION OVER FREQUENCY INTERVAL

Order reduction over given frequency intervals, low-pass, band-pass and high-pass, is considered in this section. In all three cases, only the gain of the high-order system is approximated. This is done for brevity, however, the proposed technique can be extended to approximating the system's phase in the cases of low-pass and high-pass order reduction. Band-pass approximation of phase is difficult to obtain in the ζ-plane. However, since it is usually desirable to obtain a stable and minimum phase reduced-order models, gain approximation should suffice in most cases.

Low-Pass (LP) Approximations

Consider a given $\{m/n\}$ dynamic system (6) and assume that it is stable and minimal phase. It is desired to approximate, over a given (LP) frequency interval $[0,\omega_0]$, the gain of the system, whose Darlington polynomial expansion (15) maps into a power series in the ζ-plane (16).

Let $\hat{G}(s)$ be the $\{k/p\}$, $k<p<n$, required reduced order model of $G(s)$ over the frequency interval $[0,\omega_0]$ and let $\hat{g}$ be the gain of $\hat{G}(s)$, $\hat{g}= \frac{1}{2}\ln\hat{G}(s)\hat{G}(-s)$, $s=j\omega$. Its Darlington polynomial series expansion

$\hat{g}= \sum_{i=0}^{\infty} \hat{d}_{2i}D_{2i}(\omega/\omega_0)$ maps into a power series in the ζ-plane

$$\ln\hat{R}(\zeta)\hat{R}(-\zeta) = \sum_{i=0}^{\infty} \hat{d}_{2i}\zeta^{2i} \tag{19}$$

where $\hat{R}(\zeta)$ is a $\{k/p\}$ rational function, structured similarly to $\hat{G}(s)$.

Thus, $\hat{R}(\zeta)$ is sought such that

$$\ln R(\zeta)R(-\zeta) \overset{\alpha}{\equiv} \ln\hat{R}(\zeta)\hat{R}(-\zeta) \tag{20}$$

where $\overset{\alpha}{\equiv}$ denotes the requirement that the first α terms in the series expansions of the left-hand and right-hand sides be identical. Using (16) in (20) we have

$$l n\hat{R}(\zeta)\hat{R}(-\zeta) \overset{\alpha}{\equiv} \sum_{i=0}^{\infty} d_{2i}\zeta^{2i} \tag{21}$$

Consider the relationship

$$l n \sum_{i=0}^{\infty} a_{2i}\zeta^{2i} = \sum_{i=0}^{\infty} d_{2i}\zeta^{2i} \tag{22}$$

Substituting $\zeta=0$ in (22) yields $a_0=\exp(d_0)$ and differentiating (22) with respect to ζ yields

$$a_{2i} = \frac{1}{i}\sum_{j=1}^{i} j d_{2j} a_{2(i-j)} \quad i= 1,2,\ldots \tag{23}$$

Hence, knowledge of the coefficients d_{2i} enables the derivation of the coefficients a_{2i}.

Now, substituting (22) into (21) yields the requirement

$$\sum_{i=0}^{k} \gamma_i\zeta^{2i} \Big/ \sum_{i=0}^{p} \delta_i\zeta^{2i} \overset{\alpha}{\equiv} \sum_{i=0}^{\infty} a_{2i}\zeta^{2i} \tag{24}$$

with $\delta_0 = 1$ for unique determination of the coefficients.

Note that our problem has turned out to be the classical Padé approximation problem in ζ^2.

Equating the first k+1 powers in (24) yields

$$\gamma_i = \sum_{j=0}^{i} a_{2j}\delta_{i-j} , \quad i= 0,1,\ldots,k \tag{25}$$

and equating further p powers (recall that $\delta_0=1$) yields

$$\sum_{i=1}^{p} a_{2(k+j-i)}\delta_i = a_{2(k+j)} , \quad j=1,2,\ldots,p \tag{26}$$

Solving the system of equations (26) yields the coefficients $\delta_1,\delta_2,\ldots,\delta_p$ which can be used to determine the coefficients $\gamma_0,\gamma_1,\ldots,\gamma_k$ using (25). Thus, α in (20) is determined by $\alpha = k+p+1$, namely, to determine the {k/p} Padé approximation (24), k+p+1 Darlington coefficients d_{2i} are required.

Note that since $\hat{R}(\zeta)\hat{R}(-\zeta)$ is a rational function in ζ^2, low-pass approximation of the system's gain allows the selection of reduced order models with left-hand s-plane zeros and poles.

High-Pass (HP) Approximations

Consider the given {m/n} dynamic system (6) and assume that it is stable and minimum phase. It is desired to approximate its gain over the (HP) frequency interval $[\omega_c,\infty]$. Using the transformation

$$s = 2\omega_c\Big/(\zeta-1/\zeta) , \quad |\zeta|>1 \tag{27}$$

points on the $s=j\omega$ axis, $s_\alpha \notin [j\omega_c, -j\omega_c]$, map onto the unit circle in the ζ-plane. Hence, in the ζ-plane, HP approximations are dealt with as though they were LP approximations.

Let s_i' and s_j'' , $i=1,2,\ldots,m;j=1,2,\ldots,n$, be the given system's zeros and poles respectively. Normalize each one of them to ω_c, $s \rightarrow s/\omega_c$, and transform into the inverse plane

$$p \triangleq \left(s/\omega_c\right)^{-1} = \omega_c/s \tag{28}$$

Recall the LP s-plane to ζ-plane mapping (3). Using (28), we have

$$\left(\omega_c/s\right)_{HP} = (\zeta-1/\zeta)/2 = \left(s/\omega_c\right)_{LP} \tag{29}$$

Hence, the solution of a LP approximation problem, over the frequency interval [0,1], of a transfer function whose zeros and poles are p_i' and p_i'' respectively (derived using (28)), is equivalent to solving the original HP approximation problem over the frequency interval $[\omega_c,\infty]$. Approximating in the ζ-plane results in 'p-plane' zeros and poles, thus yielding, using (28), the required approximating s-plane zeros and poles.

Thus, solving the HP model reduction problem can be done by the same algorithms used for solving the LP approximation problem. Furthermore, as in the LP model reduction case, HP approximation of the system's gain allows the selection of reduced-order models with left-hand s-plane zeros and poles.

Band-Pass (BP) Approximations

The problem of BP approximation over the frequency interval $[\omega_1,\omega_2]$, $\omega_2>\omega_1>0$, is more complex than the previous LP and HP model reduction. This is so since, in the case of BP approximation, two disjoint s-plane intervals, $[j\omega_1,j\omega_2]$ and $[-j\omega_1,-j\omega_2]$, have to be mapped onto the ζ-plane unit circle, $|\zeta|=1$, so that s-plane complex conjugate pairs would correspond to ζ-plane complex conjugate pairs.

However, if BP gain approximation only is considered, then, since the gain is an even function in s^2, the two separate intervals become one, $[\omega_1^2,\ \omega_2^2]$. Thus, the following transformation

$$s^2 = \frac{1}{4}\left(\omega_2^2-\omega_1^2\right)\left(\zeta^2+1/\zeta^2-2\right)-\omega_1^2 \quad ,\ |\zeta|>1 \tag{30}$$

yields a ζ-plane approximation problem which is similar to the LP one.

Let s_i' and s_j'', i=1,2,...,m; j=1,2,...,n, be the given system's zeros and poles respectively. Transform each one of them into

$$p^2 = \left(s^2+\omega_1^2\right)\Big/\left(\omega_2^2-\omega_1^2\right) \tag{31}$$

so that p and s are in the same quadrant, i.e, sgnRe{p} = sgnRe{s} (and then, also, $p\leftrightarrow s \Longrightarrow \bar{p}\leftrightarrow\bar{s}$). Recall the LP s-plane to ζ-plane mapping (3). Squaring it yields

$$s^2 = \omega_0^2\left(\zeta^2+1/\zeta^2-2\right)\Big/4 \tag{32}$$

Using (32) and (31), we have

$$\left[\left(s^2+\omega_1^2\right)\Big/\left(\omega_2^2-\omega_1^2\right)\right]_{HP} = \left(\zeta^2+1/\zeta^2-2\right)\Big/4 = \left(s^2/\omega_0^2\right)_{LP} \tag{33}$$

Hence, the solution of a LP approximation problem, over the frequency interval [0,1], of a transfer function whose zeros and poles are p_i' and

p_i'' respectively (derived using (31)), is equivalent to solving the original BP approximation problem over the frequency interval $[\omega_1, \omega_2]$. Approximating in the ζ-plane results in 'p-plane' zeros and poles, thus yielding, using (30), the required approximating s-plane zeros and poles.

Thus, solving the BP model reduction problem can be done using the same algorithms used for solving the LP approximation problem. Furthermore, as in the LP model reduction case, BP approximation of the system's gain allows the selection of reduced-order models with left-hand s-plane zeros and poles.

NUMERICAL EXAMPLES

Low-Pass Model Reduction : Given the {3/4} system [10]

$$G(s) = \frac{s^3+7s^2+24s+24}{s^4+10s^3+35s^2+50s+24} \tag{34}$$

To compare our approximation to order reduction by other methods, Figs. 1 and 2 depict time and frequency responses respectively of the high-order system (34), our {1/2} reduced-order model over the frequency interval [0,3] and the following approximations of (34) by other methods: (a) Davison in [10]; (b) Chidambara in [10]; (c) Shamash [10].

As can be seen from these responses, our method compares well with other procedures.

Band-Pass Model Reduction: Consider the following {0/6} system (a Butterworth band-pass filter)

$$G(s) = \frac{1.0101.10^6}{[(s+1)^2+10^2][(s+0.707)^2+10.707^2][(s+0.707)^2+9.293^2]} \tag{35}$$

Fig. 3 depicts the frequency response of this system and some of its reduced order models, {0/4}, {1/4} and {2/4}, over the frequency interval [6,10].

CONCLUSIONS

A method for reduced-order modeling of linear, time-invariant, SISO systems over desired frequency intervals was proposed. Low-pass, band-pass and high-pass approximations were considered. In all cases, model reduction was accomplished by manipulating two orthogonal polynomial series, one representing the high-order system and the other representing the approximating low-order model.

Whilst only the gain of the high-order system was approximated in this paper for all three frequency bands, the technique can be extended to approximating the system's phase, but for the case of band-pass approximation. Since it is usually desirable to obtain a stable and minimum phase reduced-order models, gain approximation with our method should suffice in most cases.

Since model reduction in the control literature is mainly concerned with low-pass systems, the only numerical comparison we could make between our method and others is by considering a low-pass system. It was shown that the proposed method compares well with other procedures whilst having the advantage of providing reduced-order models over a desired frequency interval.

REFERENCES

[1] R. Genesio and M. Milanese, "A note on the derivation and use of reduced models". IEEE Trans. Aut. Control, Vol. AC-21, pp. 118-122, 1976.

[2] C. F. Chen and L. S. Shieh, "An algebraic method for control system design". Int. J. Control, Vol. 11, pp. 717-739, 1970.

[3] M. Lal and R. Mitra, "Simplification of large system dynamics using a moment evaluation algorithm". IEEE Trans. Aut. Control, Vol. AC-19, pp. 602-603, 1974.

[4] Y. Shamash, "Stable reduced order models using Padé type approximations". Ibid., pp. 615-616, 1974.

[5] M. F. Hutton and B. Friedland, "Routh approximations for reducing order of linear time-invariant systems". Ibid., Vol. AC-20, pp. 329-337, 1975.

[6] S. C. Chuang, "A method for linear systems order reduction". IEEE Conference Publication, No. 96, pp. 213-219, 1973.

[7] M. Lal and M. E. Van Valkenburg, "Order reduction of linear dynamic models". Proc. 7th Ann. Pittsburgh Conf. on Modeling and Simulation, pp. 254-258, 1976.

[8] S. Darlington, "Network synthesis using Tchebysheff polynomial series", BSTJ, Vol. 31, pp. 613-665, 1952.

[9] Y. Bistritz and G. Langholz, "Model reduction by Chebyshev polynomials techniques". Tel-Aviv University, School of Engineering, Report TAU/SOE 459/77 (submitted for publication).

[10] Y. Shamash, "Linear systems reduction using Padé approximants to allow retention of dominant modes". Int. J. Control, Vol. 12, pp. 257-272, 1975.

APPENDIX

An n-th order Chebyshev polynomial is

$$T_n(x) = \cos(n\cos^{-1}x), \quad x\varepsilon[-1,1]$$

Darlington and Chebyshev polynomials can be shown to be related by

$$D_n(x) = j^n T_n(x) \quad , \quad n>0 \quad , \quad x\varepsilon[-1,1]$$

Darlington polynomials are recursively related

$$D_{n+1}(x) = 2jxD_n(x)+D_{n-1}(x) \quad , \quad n=1,2,\ldots$$

where $D_0(x) = 1$ and $D_1(x) = jx$.

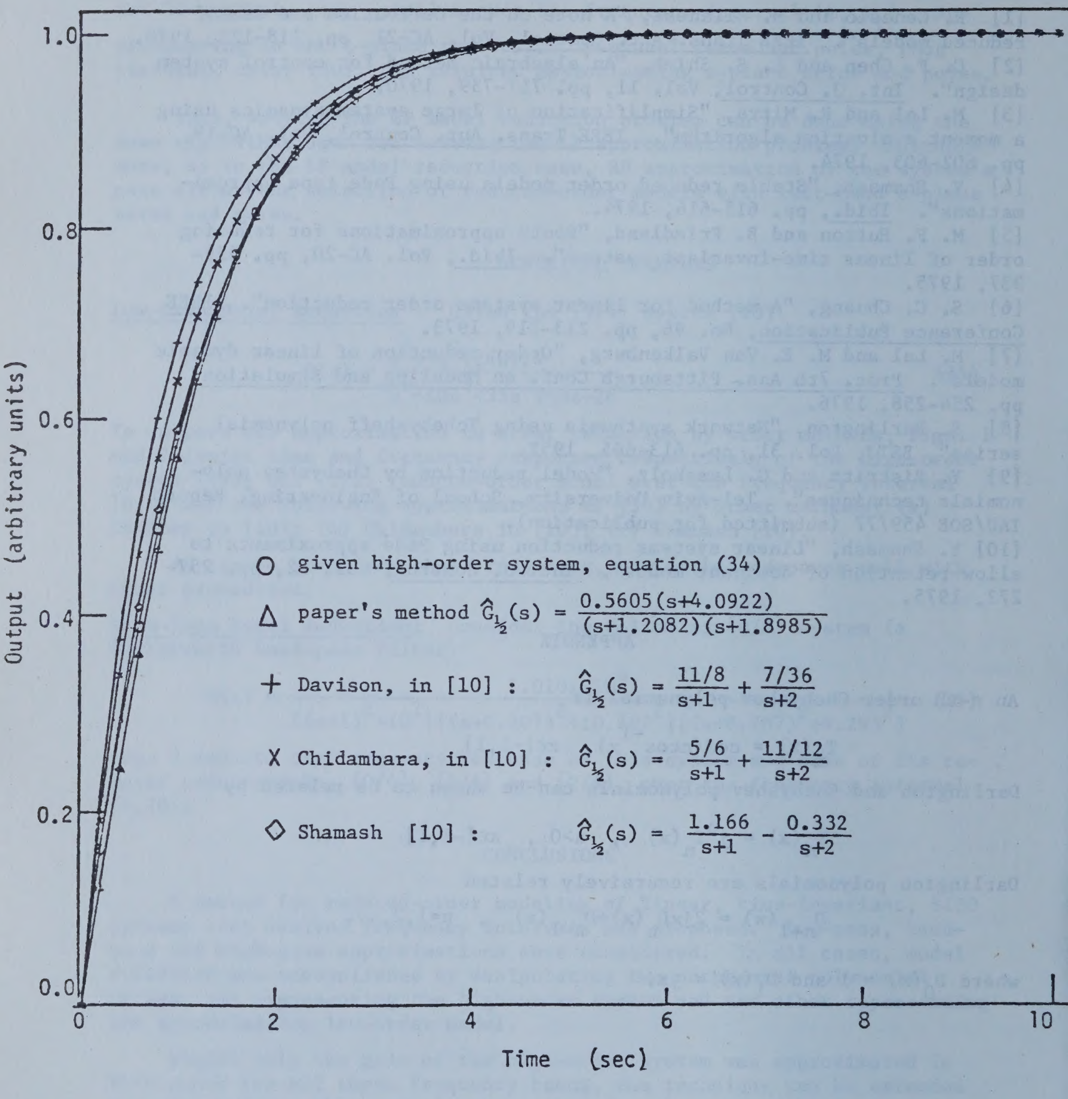


Fig. 1. Comparison of time responses to a unit step input of the given high-order system (34), its {1/2} reduced-order model over the frequency interval [0,3] and various approximations by other methods.

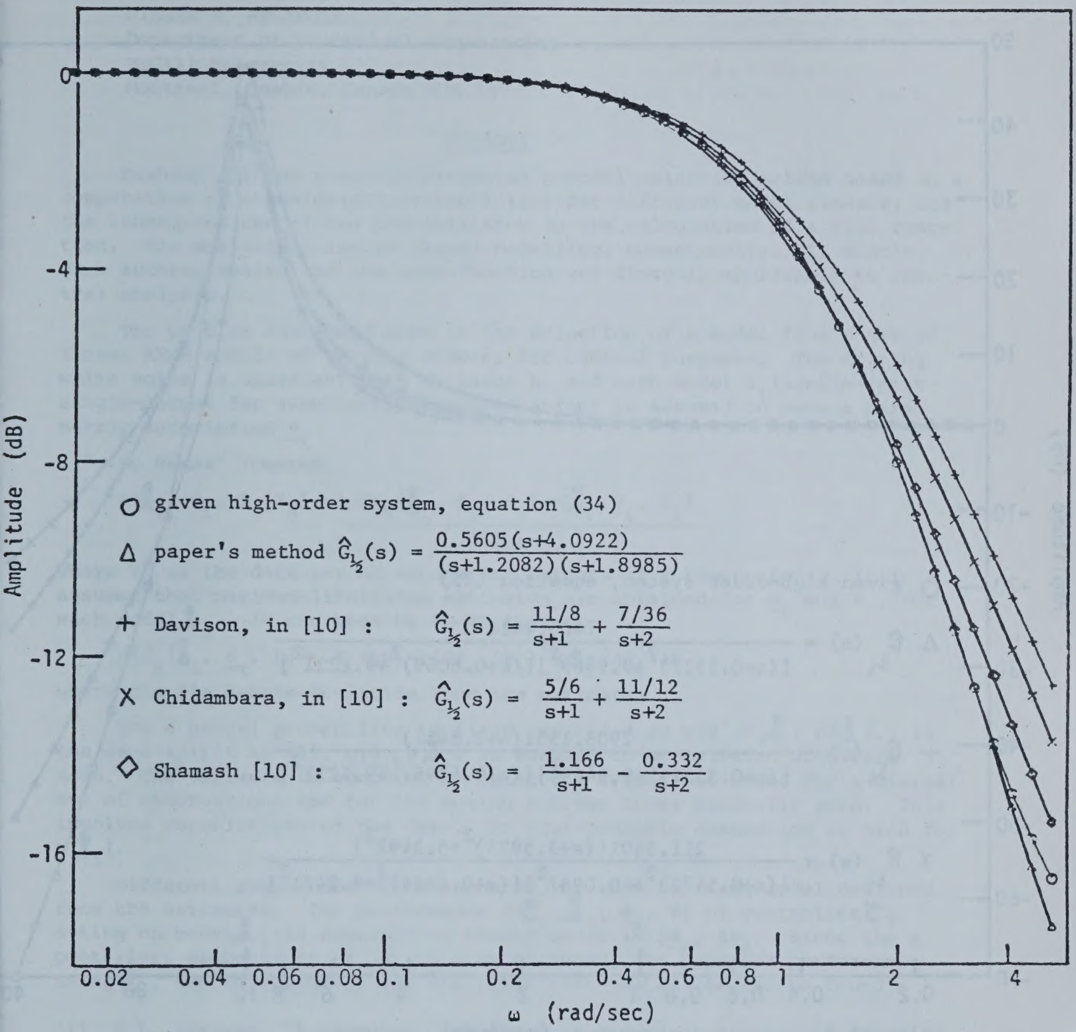


Fig. 2 Comparison of frequency responses of the given high-order system (34), its {1/2} reduced-order model over the frequency interval [0,3] and various approximations by other methods.

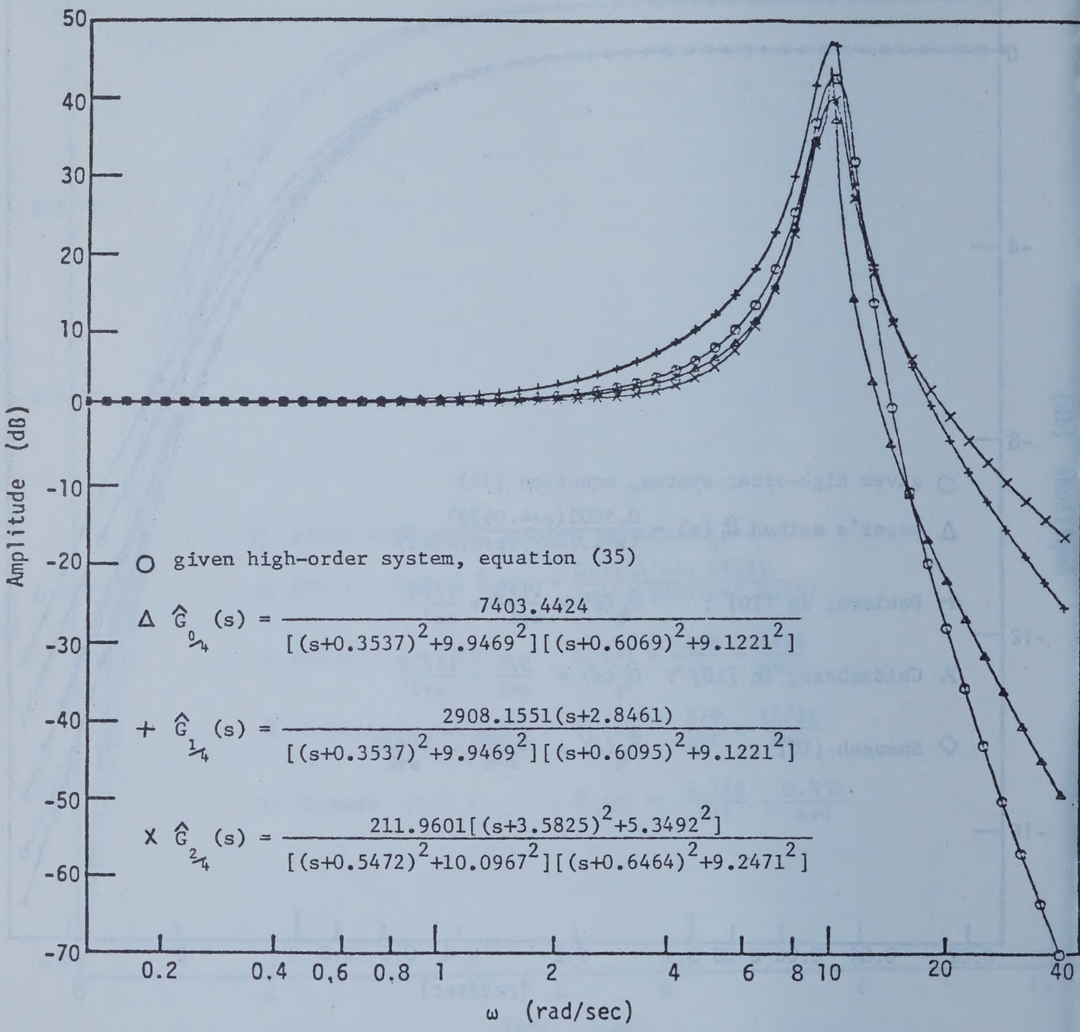


Fig. 3. Frequency response of the given high-order system (35) and some of its reduced-order models over the frequency interval [6,10].

SELECTION OF MODEL STRUCTURE FOR CONTROL

PIERRE R. BELANGER
Department of Electrical Engineering
McGill University
Montreal, Quebec, Canada H3A 2A7

SUMMARY

Kashyap [1] has recently presented a model selection method based on a computation of a posteriori probabilities for different model classes, and the subsequent use of the probabilities in the calculation of a risk function. His motivation lay in signal modelling; consequently, his models were autoregressive and the risk function was directly applicable to spectral analysis.

The problem addressed here is the selection of a model from a set of linear ARMA models of varying orders, for control purposes. The driving white noise is Gaussian, with variance R, and each model S_i (single-input-single-output for simplicity of presentation) is assumed to have a parametric description $\underline{\theta}_i$.

By Bayes' Theorem,

$$p[\mathcal{S}_i, \underline{\theta}_i, R_i | Y_N] = \frac{p[Y_N | \mathcal{S}_i, \underline{\theta}_i, R_i]\, p[\mathcal{S}_i, \underline{\theta}_i, R_i]}{p[Y_N]}$$

where Y_N is the data set of inputs and outputs at N time points. It is assumed that maximum-likelihood estimates are obtained for $\underline{\theta}_i$ and R_i, for each model $\mathcal{S}_i$. It can then be shown that

$$p[Y_N | \mathcal{S}_i, \underline{\theta}_i, R] = K_i \exp\{-\tfrac{1}{2}[\Delta\underline{\theta}_i \ \ \Delta R_i]^T N \mathbf{S}_i [\Delta\underline{\theta}_i \ \ \Delta R_i]\}$$

where the Δ's denote deviation from the estimates.

The a priori probability is first expressed as $p[\underline{\theta}_i, R | \mathcal{S}_i]\, p[\mathcal{S}_i]$. In the same spirit as [1], $p[\underline{\theta}_i, R | \mathcal{S}_i]$ is assumed to be Gaussian with zero mean. The variance is computed by using the Cramer-Rao bound for a minimal set of observations and for the moving average terms nominally zero. This involves correlations of the data. An equi-probable assumption is used for $p[\mathcal{S}_i]$.

Different regulators are considered, eg. one for each model designed from the estimates. The performance $J(\mathcal{C}_j, \mathcal{S}_i, \underline{\theta}_i, R)$ of controller $\mathcal{C}_j$ acting on model $\mathcal{S}_i$ is expanded to second order in $\Delta\underline{\theta}_i$, ΔR_i. Since the a posteriori probability of $(\mathcal{S}_i, \underline{\theta}_i, R)$ is known, the expected performance using $\mathcal{C}_j$, can be computed, and the least-risk controller can be found.

[1] R.L. Kashyap, "A Bayesian Comparison of different classes of dynamic models using empirical data", IEEE Trans. on Automatic Control, Vol. AC-22, pp. 715-727, October 1977.

A FREQUENCY DESIGN TECHNIQUE FOR MODEL MATCHING AND SENSITIVITY REDUCTION*

DESANTIS, R.M.
Département de Génie Electrique
Ecole Polytechnique
Montreal, Canada

LEFEBVRE, S.
Département de Génie Electrique
Ecole Polytechnique
Montreal, Canada

ABSTRACT

The present paper illustrates the application of compensators with a perfect matching and/or absolute invariance property to multivariable feedback design. Contrary to the traditional approach whereby these special compensators are viewed as elements to physically construct and insert into the feedback chain, they are simply viewed here as a second term of comparison to assess the performance of other compensators. This leads to a graphical technique which promises to be a useful complement to the already available frequency design procedures.

INTRODUCTION

Recent contributions to practical multivariable design techniques such as the Rosenbrock's Inverse Nyquist Array Method [1]or the Mac Farlane's Characteristic Loci Method [2]share a design philosophy whereby one first attempts to minimize the mutual interactions between the elements of a selected set of input-output pairs of the original open loop system. The compensator is subsequently conceived by ignoring the interactions and by applying the classical frequency and/or root locus scalar techniques. While these methods have been reported to work well in a number of practical situations, [1 - 4 - 7], some difficulties have been encountered in other cases [5 - 6]. As to be expected, the difficulties arise either when a preliminary decoupling is not desirable [8] , or when the residual interactions are not negligeable.

The present paper is to the effect that these difficulties can in part be alleviated by complementing the usual open loop frequency plots with the frequency responses of the classical absolute invariant and/or perfect matching compensators. These additional data facilitate the design procedure by allowing one to visualize not only the effect that a scalar compensator has on the output related to the pair on which it is directly connected, but also the effect that it will have on all the other outputs which are interconnected with this pair. The paper is a natural follow-up of [9] .

BASIC CONCEPTS AND RESULTS

Though the development concerns general linear time-invariant multivariable systems, for simplicity of exposition we will confine our discussion to a two inputs-two outputs - one disturbance system of the type represented in Figure 1. A feedback compensator configuration that one may adopt to improve the performance of this system is illustrated in Figure 2: to the original input one finds added a corrective action linearly related to the difference, $\underline{\Delta}$, between the output of the actual system and that of

---

[*] This research was supported in part by the Canadian National Research Council, grant CNRC-A8244, and by the Government of Quebec, grant FCAC-CRP-378-76.

the ideal plant. The compensator M plays an active role only when the output of the actual plant represented by the matrix transfer function G(s) is different from the output of the ideal plant B(s). Its purpose is to make the actual output look like the ideal one.

Let (u_1, u_2) represent the Laplace transform of the inputs, u_3 that of the disturbance, $(\Delta_1^{0k}, \Delta_2^{0k})$ and $(\Delta_1^{k}, \Delta_2^{k})$ those of the open and closed loop error contributions which are induced by the presence of the input, u_k, k=1,2,3. Clearly, the systems in Figures 1 and 2 can be studied in terms of the following mathematical model :

$$\underline{\Delta}^0(s) = \sum_{k=1} \Delta^{0k} = \hat{G}(s)\, \underline{u}(s)$$

$$\underline{\Delta}(s) = S(s)\, \hat{G}(s)\, \underline{u}(s) = S\, \underline{\Delta}^0(s)$$

$$\hat{G}(s) = G(s) - B(s)$$

$$S(s) = [I + G(s)\, M(s)]^{-1}$$

$$B(s) = \begin{bmatrix} B_{11}(s) & B_{12}(s) & 0 \\ B_{21}(s) & B_{22}(s) & 0 \end{bmatrix}$$

Definition: The perfect matching, PM, compensator D_k^{11}, with k=1,2,3, is defined by the condition that if $M_{11}=D_k^{11}$ and $M_{ij}=0$ for $(i,j) \neq (1,1)$ then $\Delta_2^k=0$. Similarly, the P.M. compensators D_k^{22}, D_k^{21}, and D_k^{12} are defined by the following conditions :

if $M_{22} = D_k^{22}$ and $M_{ij} = 0$ for $(i,j) \neq (2,2)$ then $\Delta_1^k = 0$

if $M_{21} = D_k^{21}$ and $M_{ij} = 0$ for $(i,j) \neq (2,1)$ then $\Delta_2^k = 0$

if $M_{12} = D_k^{12}$ and $M_{ij} = 0$ for $(i,j) \neq (1,2)$ then $\Delta_2^k = 0$

Proposition 1 (Analytic expression of the Perfect Matching Compensators). The PM compensators associated with the open loop system in Figure 1 are given by the following expressions:

$$D_k^{11} = \frac{\hat{G}_{2k}}{G_{21}\hat{G}_{1k} - \hat{G}_{2k}G_{11}} \ ; \quad D_k^{12} = \frac{\hat{G}_{1k}}{G_{11}\hat{G}_{2k} - \hat{G}_{1k}G_{21}}$$

$$D_k^{21} = \frac{\hat{G}_{2k}}{G_{22}\hat{G}_{1k} - \hat{G}_{2k}G_{12}} \ ; \quad D_k^{22} = \frac{\hat{G}_{1K}}{G_{12}\hat{G}_{2k} - \hat{G}_{1k}G_{22}}$$

Proposition 2 (Feedback structural properties of the Perfect Matching Compensators). The PM compensators D_{kc}^{ij} associated with the closed loop system in Figure 2 can be expressed in terms of the PM compensators D_k^{ij} associated with the open loop system in Figure 1 according to the following formula

$$\begin{bmatrix} D_{kc}^{11} & D_{kc}^{12} \\ D_{kc}^{21} & D_{kc}^{22} \end{bmatrix} = \begin{bmatrix} D_k^{11} & D_k^{12} & \alpha & 0 \\ D_k^{21} & D_k^{22} & 0 & \beta \end{bmatrix}$$

where

$$\alpha = 1 - \frac{M_{11}}{D^{11}} - \frac{M_{21}}{D^{21}} \ , \quad \beta = 1 - \frac{M_{22}}{D^{22}} - \frac{M_{12}}{D^{12}}$$

Note that for $k=3$ the perfect matching compensators D_3^{ij} coincide with the absolute invariant compensators considered in the classical literature of some twenty years ago [9]. Observe also that these P.M. compensators are neither necessarily well defined nor, when well defined, are they physically realizable nor do they guarantee system stability. All these shortcomings explain why these compensators which have been traditionally viewed as elements to physically construct and insert into the plant, are paid only a scant interest nowdays.

The next propositions show on the other hand that an alternative way to exploit these compensators is to consider them as a second term of comparison to assess the performance of any other given set of compensators. While this novel type of application is not hampered by the irrealizability, instability or complexity of the P.M. compensators, it also lends itself to a graphical interpretation which provides a natural extension of the classical frequency procedures to multivariable system design.

Proposition 3 (The relation between the sensitivy operator and the Perfect Matching compensators).

The sensitivity operator matrix related to the k - the input has the following expression

$$S^k = \frac{1}{\det[1+G(s)\,M(s)]} \begin{bmatrix} 1 - \frac{M_{22}}{D_k^{22}} - \frac{M_{12}}{D_k^{12}} & 0 \\ 0 & 1 - \frac{M_{11}}{D_k^{11}} - \frac{M_{21}}{D_k^{21}} \end{bmatrix}$$

(S^k is defined by $\underline{\Delta}^k = S^k \underline{\Delta}^{0k}$

Proposition 4 (Analytic expression for the error attenuation factors). The error attenuation factors $|\Delta_1^k/\Delta_1^{0k}\,(jw)$ and $\Delta_2^k/\Delta_2^{0k}\,(jw)$ can be evaluated analytically (graphically) by using the following expressions

$$\frac{\Delta_1^k}{\Delta_1^{0k}}(jw) = \frac{1}{1+M_{11}G_c^{11}} \cdot \frac{1}{1+M_{21}G_c^{12}} \cdot \frac{1+M_{12}Q_{ck}^{12}}{1+M_{12}G_c^{21}} \cdot \frac{1+M_{22}Q_{ck}^{22}}{1+M_{22}G_c^{22}}(jw)$$

$$\frac{\Delta_2^k}{\Delta_2^{0k}}(jw) = \frac{1+M_{11}Q_{ck}^{11}}{1+M_{11}G_c^{11}} \cdot \frac{1+M_{21}Q_{ck}^{21}}{1+M_{21}G_c^{12}} \cdot \frac{1}{1+M_{12}G_c^{21}} \cdot \frac{1}{1+M_{22}G_c^{22}}(jw)$$

where G_{cij} represents the transfer function between the input j and the output i of the system in which one has already imbedded the compensators appearing to the left of the term containing M_{ij} in the above sequence and where $Q_{ck}^{ij} = -1/D_{ck}^{ij}$.

To illustrate the type of graphical visualization which is naturally suggested by Proposition 4, suppose that we want to study the effect of inserting a scalar compensator M_{22} to the system in Figure 1. Then following Proposition 4 one is led to draw the Nyquist plots of the frequency responses of G_{22}, $-1/M_{22}$ and Q_k^{22}, k-1,2,3. Using these plots one can visualize the frequency behavior of $|\Delta_1^k/\Delta_1^{0k}(jw)|$ and $|\Delta_0^k/\Delta_2^{0k}(jw)|$ in terms of the vectors $-1/M_{22}(jw)$; $-1/M_{22}(jw) + G_{22}(jw)$; and $-1/M_{22}(jw) + Q_k^{22}(jw)$. A typical example is represented in Figure 3; in this particular case, for w=10, one has for instance

$$\left.\frac{\Delta_2^k}{\Delta_{02}^k}\right|_{w=10} = \left.\frac{-1/M_{22}}{-1/M_{22}+G_{22}}\right|_{w=10} = \frac{m_1}{m_4}, \quad k = 1,\ 2,\ 3$$

$$\left.\frac{\Delta_1^1}{\Delta_{01}^1}\right|_{w=10} = \left.\frac{-1/M_{22}+Q_1^{22}}{-1/M_{22}+G_{22}}\right|_{w=10} = \frac{m_3}{m_4}$$

$$\left.\frac{\Delta_1^2}{\Delta_{01}^2}\right|_{w=10} = \left.\frac{-1/M_{22}+Q_2^{22}}{-1/M_{22}+G_{22}}\right|_{w=10} = \frac{m_3}{m_4}$$

$$\left.\frac{\Delta_1^3}{\Delta_{01}^3}\right|_{w=10} = \left.\frac{-1/M_{22}+Q_3^{22}}{-1/M_{22}+G_{22}}\right|_{w=10} = \frac{m_5}{m_4}$$

A completely analogous procedure can be applied by drawing the Nyquist plots of $M_{22}(jw)$, $-1/G_{22}(jw)$ and $D_k^{22}(jw)$.

An illustrative example

The influence of small load variations over voltage and frequency deviations in a DC-AC converter of the type represented in Figure 4 can be studied in terms of the linear time invariant model represented in Figure 1, where

$$G_{11}(s) = \frac{\Delta F(s)}{\Delta U_1(s)} = \frac{.5125}{1+.55s}\, e^{-.08s}$$

$$G_{12}(s) = \frac{\Delta F(s)}{\Delta U_2(s)} = \frac{-.145}{1+.67s}\, e^{-.10s}$$

$$G_{21}(s) = \frac{\Delta V_a(s)}{\Delta U_1(s)} = \frac{1.775}{1+.41s}\, e^{-.04s}$$

$$G_{22}(s) = \frac{\Delta V_a(s)}{\Delta U_2(s)} = \frac{.675}{1+.32s} + \frac{.30s}{(1+.18s)^2}$$

$$G_{13} = \frac{\Delta F(s)}{\Delta W(s)} = \frac{-.188}{1+.6s}\, e^{-.06s}$$

$$G_{23} = \frac{\Delta V_a(s)}{\Delta W(s)} = \frac{-1.20}{1+.34s}$$

In this model U_3 corresponds to the load variation; Δ_1^o and Δ_2^o represent respectively the frquency and voltage deviations; U_1 denotes the voltage applied to the DC motor while U_2 is the voltage applied to the alternator excitation circuit.

Let this system be embedded in the feedback configuration represented in Figure 2, and let us consider the problem of determining a compensator M(s) so that the frequency and voltage of the converter follow the values of U_1 and U_2 and are insensitive to U_3. In particular let the following specifications be required. The feedback system should be stable and remain stable in the presence of any given loop failure. For $\underline{U} = U_{-1}(t)\ [1,\ 1,\ 1\ ]$ it is desired that

$$\lim_{t\to\infty} \underline{\Delta}(t) = 0 \quad \text{(perfect matching at } w=0\text{); moreover for } w \in [0,\ 2\ ]\text{, let us require that}$$

$$\frac{\Delta_2^k(iw)}{\Delta_{02}^k(iw)} < .35$$

under the constraint

$$\frac{\Delta_1^k(iw)}{\Delta_{01}^k(iw)} < 1$$

A solution of this problem via the graphical technique presented in the previous section might proceed as follows. First one draws four separate sets of Nyquist plots with each set containing the frequency response of $G_{ij}(jw)$ and $Q_k^{ij}(jw)$, k=1,2,3. From a comparative analysis of these curves one can determine that the insertion of a scalar compensator on the 2-2 loop appears to be more promising than the insertion of a scalar compensator on any one of the four possible loops. From a more detailed analysis of the Nyquist plots of $G_{22}(jw)$ and $Q_k^{22}(jw)$, (Figure 3), one finds that the insertion of the proportional + integral compensator $M_{22} = 1 + \frac{10}{s}$ provides a substantial step forward into meeting the desired specifications. At this point a second comparative analysis is performed on the four sets of Nyquist plots $G_{ijc}(s)$ and Q_{kc}^{ij} which are related to the original system modified by the insertion of $M_{22} = 1 + \frac{10}{s}$. From this analysis one finds that the application of a scalar feedback on the 1-1 loop appears to be more promising than on any other loop. From a further study of the plots of G_{11c} and Q_{kc}^{11} one easily comes to the conclusion that the addition of the proportional + integral $M_{11} = 2 + \frac{4}{s}$ allows the errors specifications to be met. Stability and integrity can be easily verified by applying classical techniques on the already available Nyquist plots (the techniques discussed in [11] may be helpful in this last step).

A comparison between the stop responses of the new system and those of the original open loop are illustrated in Figure 7.

REFERENCES

1. H.H. Rosenbrock, Computer-Aided Control System Design, Academic Press, 1974.

2. A.G.J. Mac Farlane and J.J. Belletrutti, The Characteristic Locus Design Method, Automatica, Vol. 9, pp. 575-588, 1973.

3. Decoupling the Transient Dynamics of Adjacent Alternators, J.R. Bumby and C. Priece, I.E.E.E. Trans. on PAS, Vol. PAS-94, No.4, pp. 1147-1156, July/August 1975.

4. A.K. Sinha and D.A. Rutherford, Computer-Aided Controller Design for Paper Machine, Proc. I.E.E.E., Vol. 123, No.10, pp. 1021-1025.

5. A.J. Maloney, Graphics Analysis of Dominance in Set Engine Control Models, Tech. Report of the University of Notre-Dame, No.EE-765, June 1976.

6. V. Seshadri and M.K. Sain, Multivariable System Compensation Including Characteristic Methods for Set Engine Control, Tech. Report of the University of Notre-Dame, No. EE-763, May 1976.

7. F.M. Hughes, A Simplified Frequency-Response Design Approach for Power-Station Control, Int. J. Control, Vol. 25, No. 4, pp.575-587, 1977.

8. S. Lefebvre and R.M. De Santis, Nouvelle méthode fréquentielle pour l'analyse et la synthèse des systèmes multivariables, Tech. Report of Ecole Polytechnique of Montreal, No.EP-77-R-40, October 1977.

9. R.M. De Santis and S. Lefebvre, Comparative Sensitivity and Absolute Invariant Compensators with Application to a DC-AC Converter, 14th Annual Allerton Conference, pp.964-973, October 1976.

10. O.I. Elgerd, Electric Energy Systems Theory : In Introduction, McGraw-Hill, 1971.

11. De Santis, R.M., On the Use of the Compensator Inverse Nyquist Plot for Feedback Systems Stability Analysis, 18th Mid West Symposium on Circuits and Systems, Madison, August 1976.

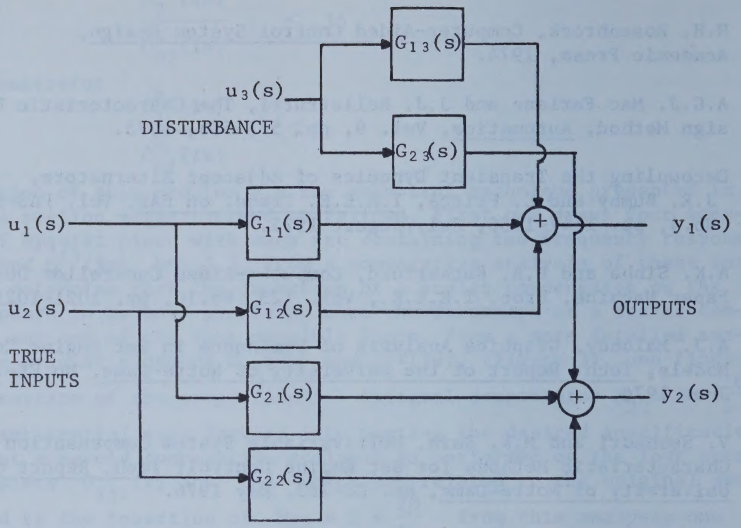

$G_{13}(s)$
$u_3(s)$
DISTURBANCE
$G_{23}(s)$
$u_1(s)$
$G_{11}(s)$
+
$y_1(s)$
$u_2(s)$
$G_{12}(s)$
OUTPUTS
TRUE INPUTS
$G_{21}(s)$
+
$y_2(s)$
$G_{22}(s)$

Figure 1 The open loop plant

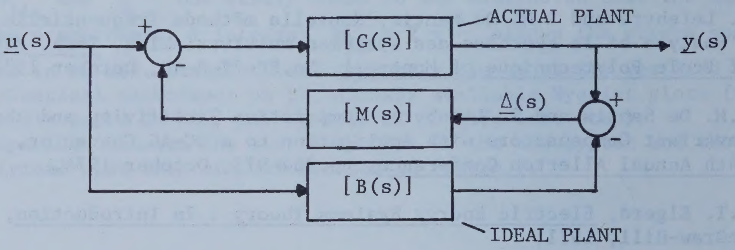

ACTUAL PLANT
$\underline{u}(s)$
+
−
$[G(s)]$
$\underline{y}(s)$
$[M(s)]$
$\underline{\Delta}(s)$
+
$[B(s)]$
IDEAL PLANT

Figure 2 The closed loop plant

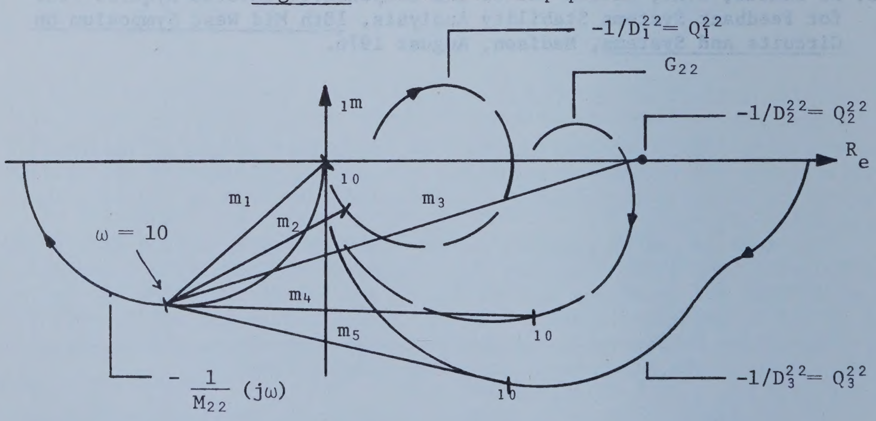
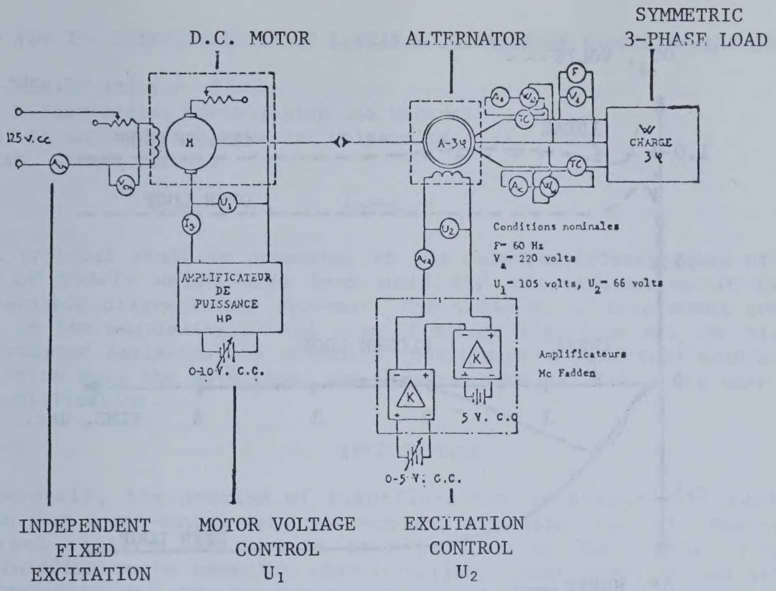


Figure 3 Illustrating the graphical use of the perfect matching compensators.


SYMMETRIC 3-PHASE LOAD
D.C. MOTOR
ALTERNATOR
125 V.cc
M
A-3φ
W
CHARGE 3φ
AMPLIFICATEUR DE PUISSANCE HP
0-10 V. C.C.
Conditions nominales
F = 60 Hz
V$_a$ = 220 volts
U$_1$ = 105 volts, U$_2$ = 66 volts
Amplificateurs Mc Fadden
K
5 V. C.C.
0-5 V. C.C.
INDEPENDENT FIXED EXCITATION
MOTOR VOLTAGE CONTROL U$_1$
EXCITATION CONTROL U$_2$

INPUTS: U_1, U_2
OUTPUTS: FREQUENCY AND VOLTAGE ON THE 3φ LINE (F; V_a)
DISTURBANCE: DETER MINISTIC VARIATION IN THE LOAD DEMAND (W)

Figure 4: Simple equivalent of an isolated electric power plant.

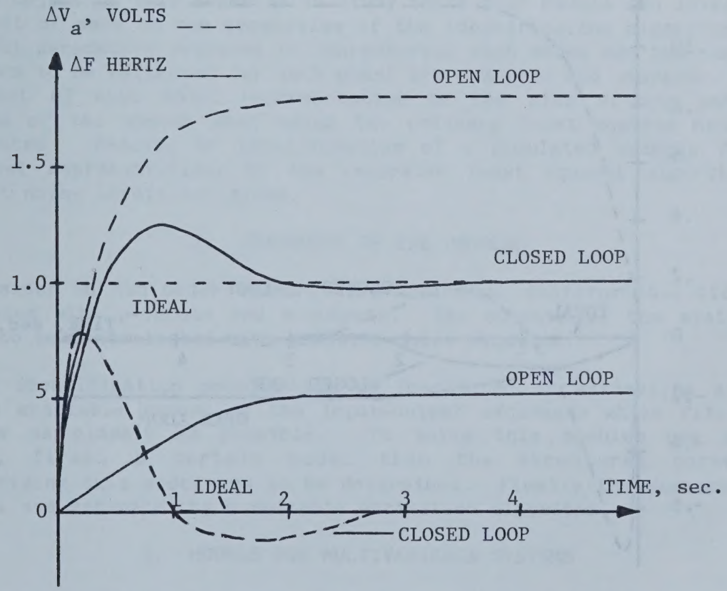


Figure 5.1 Systems responses for $U_1 = U_{-1}(t)$

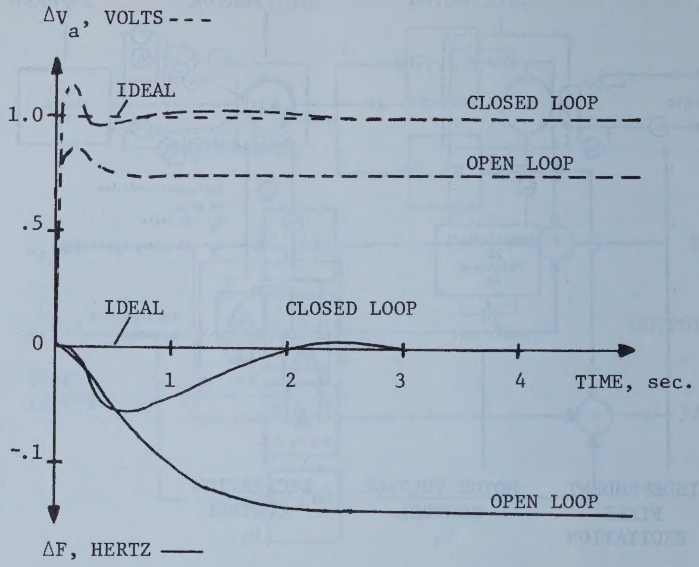


Figure 5.2 Systems responses for $U_2 = U_{-1}(t)$

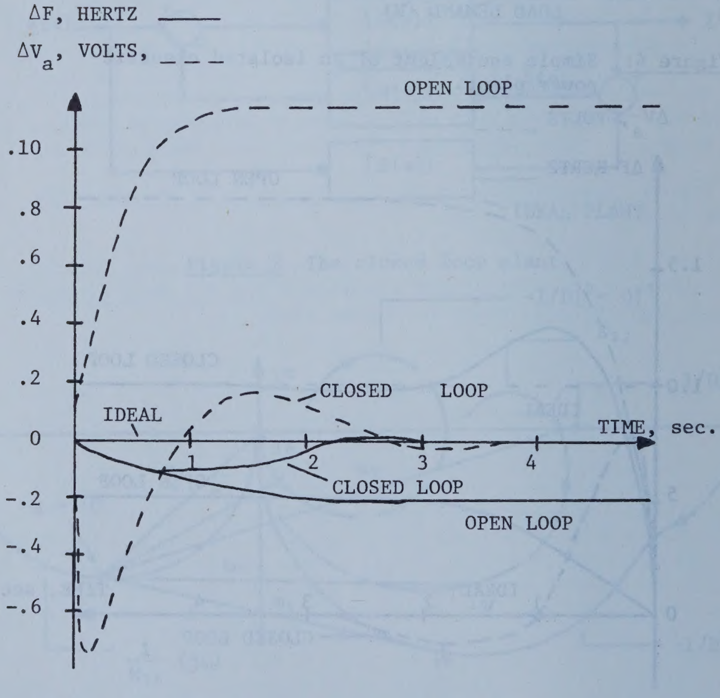


Figure 5.3 Systems responses for $U_3 = U_{-1}(t)$

MODELS FOR THE IDENTIFICATION OF LINEAR MULTIVARIABLE DISCRETE-TIME SYSTEMS

H. EL-SHERIEF and N.K. SINHA
Group on Simulation, Optimization and Control
Faculty of Engineering, McMaster University
Hamilton, Canada L8S 4L7

ABSTRACT

A critical study is presented of the relative effectiveness of four types of models which have been used for identification of linear multivariable discrete-time systems. The features of each model and its effect on the complexity of the identification algorithm and the bias of the parameter estimates are studied. Results of a simulated example are given which show the advantages and disadvantages of each model when used for identification.

1. INTRODUCTION

Recently, the problem of identification of linear multivariable systems from input-output data has received much attention [1]. One of the important factors involved in this problem is the choice of model representation to be used for identification. Each model chosen affects the complexity and the biasedness of the identification algorithm. In general, there are four types of system models which have been studied and used for identification of linear multivariable discrete-time systems. These models are: the transfer-function matrix, the impulse response matrix, the input-output difference equations and the state-space formulation. These four models are equivalent and transformations between them are possible.

The object of this paper is to study these four models and investigate the effect of each on the properties of the identification algorithm. The structural parameters required to characterize each model and the number of parameters to be estimated for each model are examined and compared. Also, the effect of each model representation on the bias of the parameter estimates of the system when using the ordinary least squares method is investigated. Results of identification of a simulated example for the four model representations by the recursive least squares algorithm for different noise levels are given.

2. STATEMENT OF THE PROBLEM

Consider an nth order linear, time-invariant, multivariable discrete-time system with p-inputs and m-outputs. The outputs of the system are assumed to be contaminated with additive noise sequence.

The identification problem can be formulated as estimating a model from the available record of the input-output sequences which fits these sequences as closely as possible. To solve this problem one has to specify, first, a certain model then the structural parameters characterizing this model are to be determined. Finally the parameters of the model are estimated by a suitable estimation algorithm.

3. MODELS FOR MULTIVARIABLE SYSTEMS

In this section the four different types of models for linear multivariable systems will be studied.

3.1 Transfer-function Matrix

The linear discrete-time system can be represented by

$$y(k) = G(z)\ u(k) \tag{1}$$

where u(k) is the p-dimensional input vector sequence, y(k) is the m-dimensional output vector sequence, G(z) is the transfer-function matrix of the system and z is the unit advance operator. Different forms of the transfer-function matrix have been used for identification. The following general form for G(z),

$$G(z) = \left[\frac{A_{ij}(z)}{B_{ij}(z)}\right] \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, p \tag{2}$$

where $A_{ij}(z)$ and $B_{ij}(z)$ are polynomials in z of degree less than or equal to n, has been considered for identification by Abaza [2] and Sinha and Caines [3]. Another form for G(z),

$$G(z) = \left[\frac{A^{*}_{ij}(z)}{D_{i}(z)}\right], \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, p \tag{3}$$

where $D_i(z)$ is defined as the least common denominator of the ith row of G(z) of equation (2) having the degree n_i (less than or equal to n) and $A^{*}_{ij}(z)$'s are polynomials in z of maximum degree n_i-1, has been considered for identification by Mital and Chen [4]. A third form for G(z),

$$G(z) = \frac{1}{D(z)}\left[A^{'}_{ij}(z)\right] \tag{4}$$

where D(z) is the characteristic polynomial of the system (of degree n) defined as the least common monic denominator of all minors of G(z) of equation (2) and $A^{'}_{ij}(z)$'s are polynomials in z of maximum degree n-1, has been considered for identification by Sen and Sinha [5]. The structural parameters required to characterize G(z) of equation (2) are the orders of the numerator and the denominator of each entry of G(z) and the number of parameters of the model to be estimated depends on these orders. The structural parameters for G(z) of equation (3) are the orders n_i's of each row of G(z) and the number of parameters to be estimated is $\Sigma^{m}_{i=1} n_i(p+1)$. The structural parameter for G(z) of equation (4) is the order of the system n and the number of parameters to be estimated is n(pm+1).

3.2 Impulse Response

The linear discrete-time system can also be represented by the following infinite series.

$$y(k) = [J_0\, z^{-1} + J_1\, z^{-2} + J_2\, z^{-3} + \ldots]\ u(k) \tag{5}$$

where J_0, J_1, ... are constant matrices called the Markov parameters of the system. These Markov parameters define the impulse response sequence for the system, hence it is possible to truncate the series after ℓ terms if the system is stable. The system can, therefore, be represented by

$$y(k) = [J_0\, z^{-1} + J_1\, z^{-2} + \ldots + J_{\ell-1}\, z^{-\ell}]\ u(k) \tag{6}$$

The model of equation (6) has been considered for identification by Mehra [6], El-Sherief and Sinha [7] and Sinha et. al. [8]. The structural parameter required to characterize the system in the model of equation (6) is the value ℓ, i.e. the minimum number of Markov parameters required to describe the system completely. The number of parameters of the model required to be estimated depends on the choice of the value of ℓ and is given by ℓmp.

3.3 Input-output Difference Equation

The linear discrete-time system can be represented by the following difference equation

$$P(z^{-1})\, y(k) = Q(z^{-1})\, u(k) \tag{7}$$

where
$$P(z^{-1}) = I + P_1 z^{-1} + \dots + P_{m_1} z^{-m_1} \tag{8}$$

$$Q(z^{-1}) = Q_1 z^{-1} + Q_2 z^{-2} + \dots + Q_{m_2} z^{-m_2} \tag{9}$$

and P_i's and Q_i's are constant matrices of proper dimension. The form of the matrices $P(z^{-1})$ and $Q(z^{-1})$ of equation (8) and (9) has been considered for identification by Kashyap and Nasburg [9]. A canonical input-output difference equation representation has been considered for identification by Guidorzi [10]. This representation has the following form

$$P(z) = [p_{ij}(z)], \quad i,j = 1, 2, \dots, m \tag{10}$$

$$Q(z) = [q_{ij}(z)], \quad i = 1, 2, \dots, m; \; j = 1, 2, \dots, p \tag{11}$$

where $p_{ii}(z)$ and $q_{ij}(z)$ are polynomials in z of the following form

$$p_{ii}(z) = z^{n_i} + a_{ii}(1)\, z^{n_i-1} + \dots + a_{ii}(n_i) \tag{12}$$

$$p_{ij}(z) = a_{ij}(1)\, z^{n_i-1} + \dots + a_{ij}(n_i) \tag{13}$$

$$q_{ij}(z) = b_{ij}(1)\, z^{n_i-1} + \dots + b_{ij}(n_i) \tag{14}$$

and n_i's are the observability subindices of the system [10]. The canonical form of equations (10)-(14) has been also considered for identification by Sinha and Kwong [11] and El-Sherief and Sinha [12].

The structural parameters required to characterize the system model of equations (8) and (9) are the orders m_1 and m_2 of the matrices $P(z^{-1})$ and $Q(z^{-1})$ and the number of parameters to be estimated is $m_1 p^2 + m_2 mp$. The structural parameters characterizing the model of equations (10)-(14) are the observability subindices of the system and the number of parameters to be estimated is $n(p+m)$.

3.4 State-space

The state-space representation is given by the equations

$$\begin{aligned} x(k+1) &= A\,x(k) + B\,u(k) \\ y(k) &= C\,x(k) \end{aligned} \qquad (15)$$

The form of matrices A, B and C of equation (15) is not unique and because of this nonuniqueness several canonical forms have been proposed for the purpose of identification. All of these canonical forms aim to transform the parametric model of equation (15) into a nonparametric form suitable for identification and reduce the number of parameters to be estimated. Many algorithms have been proposed to identify the system in the state-space representation from input-output data. Most of these algorithms start by identifying the system in one of the three models, transfer-function matrix [13], impulse response [14] or input-output difference equation [10], then the state-space representation is obtained by different transformations. Guidorzi [10] has obtained a unique relation between the state-space representation in a certain canonical form and the input-output difference equation [equations (10)-(14)]. The matrices $\bar{A}$ and $\bar{C}$ in this canonical form have the following structure

$$\bar{A} = [\bar{A}_{ij}]$$

$$\bar{A}_{ii} = \begin{bmatrix} 0 & & \\ \vdots & I_{n_i-1} & \\ a_{ii}(1) & a_{ii}(2) \;\ldots & a_{ii}(n_i) \end{bmatrix}, \quad \bar{A}_{ij} = \begin{bmatrix} 0 & \ldots\ldots\ldots & 0 \\ \vdots & & \vdots \\ a_{ij}(1) & a_{ij}(2) \;\ldots & a_{ij}(n_j) \end{bmatrix} \qquad (16)$$

where $\bar{C}$ consists of unit vectors, n_i's are the observability subindices and the matrix $\bar{B}$ has no special form. Guidorzi [10] has showed that the parameters of the matrix P(z) of equation (10) are the same as those of the matrix $\bar{A}$ and the parameters of the matrix Q(z) of equation (11) are obtained from those of the matrices $\bar{A}$ and $\bar{B}$. Other state-space canonical forms have been proposed for identification, e.g. [15]-[17]. The structural parameter required to characterize the model of equation (15) is the order of the system n and the number of parameters of the model to be estimated is n(n+p+m). It is known that not all of these parameters are required to specify the system model and some of them can be put to zero or one by transforming the system matrices into special canonical forms; e.g. equation (16). The structural parameters required to characterize the state-space representation of equation (16) are the observability subindices n_i's and the number of parameters to be estimated is, in general, n(p+m).

4. EFFECT OF MODEL CHOICE ON THE BIAS OF THE PARAMETER ESTIMATES

Assume that the outputs of the system are measured with additive zero-mean white noise sequence v(k),

$$z_i(k) = y_i(k) + v_i(k) \qquad i = 1, 2, \ldots, m \qquad (17)$$

where $z_i(k)$ is the ith noisy output and $v_i(k)$ is the noise at the ith output. Consider the following discrete-time observation representation which can be obtained for the ith output of the system

$$z_i(k) = U_i^T(k)\ \theta_i + e_i(k) \tag{18}$$

where $U_i(k)$ is a vector of forcing functions, θ_i is a vector of parameters to be estimated characterizing the ith output and $e_i(k)$ is the residual error in the ith observed output. Using the ordinary least squares method unbiased estimate of the parameter vector θ_i can be obtained if and only if the following two conditions are satisfied

a) The sequences $U_i(k)$ and $e_i(k)$ are statistically independent
b) The residual error sequence $e_i(k)$ has a zero-mean.

Thus bias is often introduced into parameter estimates by choosing a model which causes the forcing function to include observations which are correlated with the residual error in the measured output.

Next, the effect of choice of model representation on the bias of the parameter estimates of the model from noisy data using the ordinary least squares method will be discussed.

4.1 Transfer-function Matrix Representation

Consider the system in the transfer-function matrix form of equation (3) with the noise equation (17). The ith measured output of the system can be written as

$$z_i(k) = \sum_{j=1}^{p} \sum_{\ell=1}^{n_i} a_{ij}(\ell)\ u_j(k-\ell) - \sum_{\ell=1}^{n_i} d_i(\ell)\ z_i(k-\ell) + \sum_{\ell=1}^{n_i} d_i(\ell)\ v_i(k-\ell) + v_i(k) \tag{19}$$

where $a_{ij}(\ell)$ and $d_i(\ell)$ are the parameters of the polynomials $A^*_{ij}(z)$ and $D_i(z)$ respectively.

Equation (19) can be put into the form of equation (18) where

$$U_i(k) \triangleq [u_1(k-1)\ \dots\ u_1(k-n_i)u_2(k-1)\ \dots\ u_p(k-n_i)z_i(k-1)\ \dots\ z_i(k-n_i)]^T \tag{20}$$

$$e_i(k) \triangleq \sum_{\ell=1}^{n_i} d_i(\ell)\ v_i(k-\ell) + v_i(k) \tag{21}$$

From equations (20) and (21) we can notice that the vector sequence $U_i(k)$ is correlated with the residual error $e_i(k)$. Hence condition (a) is not satisified and the estimate of the vector θ_i is biased.

4.2 Impulse Response Representation

Consider the impulse response representation of equation (6) with the noise equation (17). The ith measured output of the system is written as

$$z_i(k) = \sum_{j=0}^{\ell-1} J_{j,i}\ u(k-j-1) + v_i(k) \tag{22}$$

where

$$J_s = [J_{s,1}^T \quad J_{s,2}^T \quad \dots \quad J_{s,p}^T]^T \tag{23}$$

Equation (22) can be put into the form of equation (18) where

$$U_i(k) \triangleq [u^T(k-1) \; u^T(k-2) \; \dots \; u^T(k-\ell)]^T \tag{24}$$

$$e_i(k) \triangleq v_i(k) \tag{25}$$

From equations (24) and (25) we can see that the vector sequence $U_i(k)$ is uncorrelated with the residual error $e_i(k)$, hence condition (a) is satisfied. Moreover, if the sequence $v_i(k)$ has zero mean then condition (b) is also satisfied. Therefore the estimate of the parameters of the impulse response model representation of the system is unbiased.

4.3 Input-Output Difference Equation Representation

Consider the input-output difference equation (7)-(9) with the noise equation (17). The ith measured output of the system can be written as

$$z_i(k) = \sum_{j=1}^{m_2} Q_{j,i} \, u(k-j) - \sum_{j=1}^{m_1} P_{j,i} \, z(k-j) + \sum_{j=1}^{m_1} P_{j,i} \, v(k-j) + v_i(k) \tag{26}$$

where $P_{j,i}$ and $Q_{j,i}$ are the ith rows of P_j and Q_j respectively.

Equation (26) can be put into the form of equation (18) where

$$U_i(k) \triangleq [u^T(k-1) \; \dots \; u^T(k-m_2) \; z^T(k-1) \; \dots \; z^T(k-m_1)]^T \tag{27}$$

$$e_i(k) \triangleq \sum_{j=1}^{m_1} P_{j,i} \, v(k-j) + v_i(k) \tag{28}$$

From equations (27) and (28) we notice that the vector sequence $U_i(k)$ is correlated with the residual $e_i(k)$ hence condition (a) is not satisfied and the estimate of the vector θ_i is biased.

4.4 State-Space Representation

Consider the state-space representation of equations (15) and (16) with the noise equation (17). The ith measured output can be written as follows (this is obtained using the relation between equation (16) and equations (10)-(14))

$$\begin{aligned} z_i(k) = & \sum_{j=1}^{p} \sum_{\ell=1}^{n_i} a_{ij}(\ell) \, u_j(k-\ell) - \sum_{j=1}^{m} \sum_{\ell=1}^{n_i} b_{ij}(\ell) \, z_j(k-\ell) + \\ & + \sum_{j=1}^{m} \sum_{\ell=1}^{n_i} b_{ij}(\ell) \, v_j(k-\ell) + v_i(k) \end{aligned} \tag{29}$$

Equation (29) can be put into the form of equation (18) where

$$U_i(k) \triangleq [u_1(k-1) \ldots u_p(k-n_i)\; z_1(k-1) \ldots z_1(k-n_i) \ldots z_m(k-n_j)]^T \tag{30}$$

$$e_i(k) \triangleq \sum_{j=1}^{m} \sum_{\ell=1}^{n_i} b_{ij}(\ell)\, v_j(k-\ell) + v_i(k) \tag{31}$$

From equations (30) and (31) we can notice that the vector sequence $U_i(k)$ is correlated with the residual error $e_i(k)$, hence condition (a) is not satisfied and the estimate of the parameters obtained is biased.

The authors have found a way of representing the state-space model in a nonparametric model form which causes the residual error to be uncorrelated with the forcing function. This is obtained as follows; using the canonical structure of the matrices $\bar{A}$ and $\bar{C}$ of equation (16) with the noise equation (17), the ith measured output of equation (15) can be expressed as

$$z_i(k) = \sum_{j=1}^{n_i} b^{n_1+\ldots+n_{i-1}+j}\, u(k-j) + \sum_{j=1}^{m} \sum_{\ell=1}^{n_j} a_{ij}(\ell)\, x_{n_1+\ldots+n_{j-1}+\ell}(k-n_i) + v_i(k) \tag{32}$$

where b^i is the ith row of B.

Equation (32) can be put into the form of equation (18) where

$$U_i(k) \overset{\Delta}{=} [u^T(k-1)\; u^T(k-2) \ldots u^T(k-n_i)\; x^T(k-n_i)]^T, \tag{33}$$

$$e_i(k) \overset{\Delta}{=} v_i(k) \tag{34}$$

From equations (33) and (34) we can see that the vector sequence $U_i(k)$ is uncorrelated with the residual error $e_i(k)$, hence condition (a) is satisfied. Moreover, if the noise sequence $v_i(k)$ has zero-mean then condition (b) is also satisfied and the parameter estimates obtained in this model form are unbiased. It can be noticed that the vector $U_i(k)$ defined in equation (33) contains the states of the system which are unknown. This problem has been solved by estimating the states and the parameters of the system in a bootstrap manner.

5. RESULTS OF SIMULATION

In this section the ideas discussed in sections 3 and 4 are demonstrated by estimating the parameters of a simulated 2-input 2-output 4th order system in the four different model representations. The given system, in the transfer-function matrix form of equation (2), is as follows

$$G(z) = \begin{bmatrix} \dfrac{1}{z-.25} & \dfrac{2}{z-.4} \\[2ex] \dfrac{2}{z-.25} & \dfrac{4}{(z-.5)^2} \end{bmatrix} \tag{35}$$

The first four Markov parameters of the system are

$$J_0 = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix},\ J_1 = \begin{bmatrix} .25 & .8 \\ .50 & 4 \end{bmatrix},\ J_2 = \begin{bmatrix} .0625 & .32 \\ .1250 & 4 \end{bmatrix},\ J_3 = \begin{bmatrix} .0156 & .1280 \\ .0313 & 3 \end{bmatrix} \qquad (36)$$

The input-output difference equation of the given system in the form of equations (10)-(14) is as follows

$$P(z) = \begin{bmatrix} z^2-.65z+.1 & 0 \\ (5/6)z-(1/3) & z^2-z+.25 \end{bmatrix},\ Q(z) = \begin{bmatrix} z-.4 & 2z-.5 \\ 2z-(2/3) & 17/3 \end{bmatrix} \qquad (37)$$

The state-space representation of this system in the canonical form of equation (16) is as follows

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -.1 & .65 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1/3 & -5/6 & -.25 & 1 \end{bmatrix},\ B = \begin{bmatrix} 1 & 2 \\ .25 & .8 \\ 2 & 0 \\ .5 & 4 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (38)$$

The above system was simulated on a CDC-6400 computer using the state-space model representation of equation (38) with zero-initial states and the two inputs were taken as uncorrelated zero-mean white noise sequences with standard deviation of value 1. To each output a zero-mean white noise sequence was added with standard deviation adjusted to vary the noise level at that output. The parameters of the above system in the four different model representations were estimated using recursive least squares method for different noise levels. A comparison of the square error of the estimated parameters after 500 iterations is shown in Table 1.

Table 1: Comparison of the square error of the estimated parameters

Model	Noise free	$\sigma_1=\sigma_2=.01$	$\sigma_1=\sigma_2=.05$	$\sigma_1=\sigma_2=.1$
Transfer-function	.0000	.0407	1.3793	3.3167
Impulse response	.0164	.0174	.0191	.0214
Input-output	.0001	6.7243	8.2693	8.5007
State-space	.0000	1.3402	1.6681	1.6919

We can see from Table 1 that small square errors of the estimated parameters have been obtained for the noise free case for all the four models. For the noisy case the square error of the parameter estimates is very small for the impulse response model but on the other hand is large for the other three models.

6. CONCLUDING REMARKS

Four different model representations have been used in the area of identification of linear multivariable systems and all these models are equivalent and transformations between them are possible. Each model has its own structural parameters which have to be determined in advance before parameter estimation.

Out of the four models used for identification the state-space model is used much because of the smaller number of parameters needed in the model when canonical forms are used and also because of its practical use in control theory. In general to identify the system in state-space form a nonparametric model for the system has to be estimated first, then estimates of the parameters of the state-space model are obtained by a certain transformation.

When estimating the parameters of the system from noisy data using the ordinary least squares method the parameter estimates are biased if the residual error is correlated with the forcing function. In general most of the identification algorithms concentrate mainly on the problem of removing this bias in the estimated parameters [e.g., generalized least squares and maximum likelihood]. It has been shown in section 4 that identifying the system in the impulse response form has an advantage over the other three forms because it results in unbiased estimate of the parameters of the system. This fact has been demonstrated by the results of simulation given in section 5.

7. REFERENCES

[1] El-Sherief, H. and Sinha, N.K., "Identification of multivariable systems: A survey", Proc. 21st Midwest Symposium on Circuits and Systems, pp. 56-62, (Ames, Iowa), 1977.

[2] Abaza, B.A., "Multivariable parameter estimation using a 2-step least squares estimator", Elect. Lett., No. 13, pp. 331-333, 1976.

[3] Sinha, S. and Caines, P.E., "On the use of register sequences as instrumental variables for the recursive identification of multivariable linear systems", Int. J. of Syst. Science, vol. 8, pp. 1041-1055, 1977.

[4] Mital, D.P. and Chen, C.T., "Identification of multivariable systems from input-output data", Proc. of Int. Conf. on System, Man and Cyb., pp. 422-426, 1974.

[5] Sen, A. and Sinha, N.K., "On-line estimation of the parameters of a multivariable system using matrix pseudo-inverse", Int. J. of Syst. Science, vol. 7, pp. 461-471, 1976.

[6] Mehra, R.K., "On-line identification of linear dynamic systems with applications to Kalman filtering", IEEE Trans. on Aut. Control, vol. AC-16, pp. 12-21, 1971.

[7] El-Sherief, H. and Sinha, N.K., "A nonparametric normalized stochastic approximation algorithm for on-line identification of multivariable systems", To appear in Int. J. of Syst. Science (1979).

[8] Sinha, N.K., Mahalanabis, A.K. and El-Sherief, H., "A nonparametric approach to the identification of linear multivariable systems", Int. J. of Syst. Science, vol. 9, pp. 420-430, 1978.

[9] Kashyap, R.L. and Nasburg, R.E., "Parameter estimation in multivariable stochastic difference equations", IEEE Trans. Aut. Control, vol. AC-19, pp. 784-797, 1974.

[10] Guidorzi, R., "Canonical structures in the identification of multivariable systems", Automatica, vol. 11, pp. 361-374, 1975.

[11] Sinha, N.K. and Kwong, Y.H., "Recursive identification of the parameters of multivariable systems", Proc. of IFAC Conf. on Multivariable Tech. Syst., pp. 323-327, Canada, 1977.

[12] El-Sherief, H. and Sinha, N.K., "Algorithm for identification of multivariable systems combining stochastic approximation and pseudo-inverse", To appear in Automatic Control Theory and Applications (1979).

[13] Furuta, K., "An application of realization theory to identification of multivariable process", Proc. of 3rd IFAC Symp. on Identification and System Parameter Estimation, pp. 939-941, (The Hague), 1973.

[14] Lobbia, R.N. and Saridis, G.N., "Identification and control of multivariable stochastic discrete systems", J. Cybern., pp. 40-59, 1973.

[15] Weinert, H. and Anton, J., "Canonical forms for multivariable system identification", Proc. of IEEE Conf. on Decision and Control, pp. 37-39, 1972.

[16] Tse, E. and Weinert, H., "Structure determination and parameter identification for multivariable stochastic linear systems", IEEE Trans. on Aut. Control, vol. AC-20, pp. 603-613, 1975.

[17] Bingulac, S.P. and Farias, M.A.C., "Identification and minimal realization of multivariable systems", Proc. of IFAC Conf. on Multivariable Tech. Syst., pp. 373-377, Canada, 1977.

AN APPLICATION OF SALUT APPROACH TO THE IDENTIFICATION OF A TWO AREA POWER SYSTEM *

LEFEBVRE, S., DESANTIS, R.M.
Département de Génie Electrique
Ecole Polytechnique
Montreal, Quebec

SALUT, G., AGUILAR-MARTIN, J.,
Laboratoire d'Automatique et
d'Analyse des Systèmes
Toulouse, France

ABSTRACT

A numerical application of the approach recently proposed by Salut to the identification of a linear time invariant multivariable system is presented. The application example consists of a two areas interconnected power system; the tool is linear quadratic filtering theory of the Kalman-Bucy type. Though the preliminary results are encouraging, further work is needed to make the algorithm operational in a real time fashion as well as to compare its performance in relation with other identification schemes.

INTRODUCTION

In what follows we illustrate a numerical application of the recently proposed Salut approach [1], to the identification of the input-output transfer matrix associated with the linear multivariable system represented in Figure 1. This system models the power-frequency interplay in a two areas interconnected power system of the type represented in Figure 2 (see [2] for more details). We will assume to have a direct access to u_1 and u_2 (the system inputs) and y_1 , y_2 (the system outputs); β_1 and β_2 represent zero mean white measurement noise with a known covariance.

THE METHOD

Consider a multivariable time invariant system represented by the operator form

$$A(D)\ y(t) = B(D)\ u(t) + C(D)\ w(t) \qquad (1)$$

where : $D = \frac{d}{dt}$ represents the differential operator;

A, B and C are polynomial matrices;

the vector function y, u and w represent the output, the input and the noise acting on the system.

---

* This research was partially supported by the Quebec Provincial Ministry of Education, Grant FCAC-378-76, and by the Canadian Engineering and Applied Sciences Research Council, Grant A-8244.

Following Salut [1] , observe that (1) admits the canonical state realization

$$\begin{aligned} \dot{x}(t) &= E_{*}x(t) - A_{*}y(t) + B_{*}u(t) + Q_{*}w(t) \\ y(t) &= E_{o}x(t) - A_{o}y(t) + B_{o}u(t) + Q_{o}w(t) \end{aligned} \qquad (2)$$

where the coefficients of the matrices E_{*} and E_{o} are either 0 or 1; their value is determined by the value of the dimension of the state, n, and of the observability indices, n_i , i = 1, 2, ... p , (p=dimension of y), which are associated with a completely controllable and completely observable state realization of (1). The coeffcients of the matrices A_{*}, B_{*}, Q_{*}, A_{o}, B_{o} and Q_{o} can be readily obtained in terms of the coefficients of the polynomial matrices A(D), B(D) and C(D), and vice versa. The missing details of this coincise and qualitative discussion can be found in [1] and [3].

If n and n_i are assumed to be known a priori, then one can obtain a least square estimate of the coefficients of A(D), B(D) and C(D) from a least square estimate of the entries of the matrices in (2). The main advantage of such an approach being that, under noise appropriate Gaussian conditions, the later estimation problem can be formulated in terms of a linear least square state estimation problem. As a consequence its solution can benefit from the classical Kalman-Bucy observer theory [4] .

This can intuitively be seen by considering the augmented state vector $\underline{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, where $x_1 = x$ is the original state vector of (2); the entries of x_2 are given by the unknown coefficients of A_{*}, B_{*}, A_{o} and B_{o}. Using this augmented state, (2) can be rewritten as follows

$$\dot{\underline{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = [\,F\,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} Q_{*} \\ 0 \end{bmatrix} w \qquad (3)$$

$$y = [\,H\,] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + Q_{o}w$$

where : $[\,F\,] = \begin{bmatrix} E_{*} & F_{12}(t) \\ 0 & 0 \end{bmatrix}$; $[\,H\,] = [\,E_{o} \quad H_{12}]$

with $F_{12}(t)$ and $H_{12}(t)$ easily computed in terms of u(t) and y(t); E_{*} and E_{o} are determined by n and n_i . Clearly, a least square estimate of the unknown parameters of (1) can now be viewed as a least square estimate of the state of (3). For the purposes of the present discussion the noise w will be supposed to be Gaussian white and with a covariance operator given by the identity matrix ; Q_{*} and Q_{o} will be assumed to be known a priori.

From [1], this state estimation problem can be solved by using the following linear optimal observer :

$$\dot{\hat{\underline{x}}}(t) = F(t)\dot{\hat{\underline{x}}}(t) + K(t)\,[y(t) - H(t)\hat{\underline{x}}(t)]$$

where

$$K(t) = (P(t)\,H^T + Q_* Q_o^{\,T})\,[Q_o Q_o^{\,T}]^{-1}$$

$$\dot{P}(t) = F(t)\,P(t) + P(t)\,F(t)^T + Q_* Q_*^{\,T}$$

$$- [P(t)\,H(t)^T + Q_* Q_o^{\,T}]^T\,[Q_o Q_o^{\,T}]^{-1}\,[P(t)H(t)^T + Q_* Q_o^{\,T}]$$

with

$\hat{\underline{x}}(t_o)$ = expected value of $\underline{x}(t_o)$

$P(t_o)$ = covariance matrice associated with $\underline{x}(t_o)$.

THE APPLICATION

The identification scheme presented in the above section has been programmed in Fortran H using the IBM-370 of the Computer Center at the Ecole Polytechnique. Exhaustive details concerning this program can be found in [3] ; a number of application examples can be found in [4] . To give an idea about how the program works we have simulated the system in Figure 1 using the CSMP package. We have subsequently fed the imposed u-values and the computed y values to our identification program which has reconstructed the input-output transfer function of the system.

With zero initial conditions we have let

$$u = u_{-1}\ (t) \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

$$\beta = \begin{matrix} \beta_1 \\ \beta_2 \end{matrix} .$$

where $u_{-1}(t)$ is the unitary step while λ_i and β_i are zero mean white noise with covariance $\sigma_{\lambda_1} = \rho_{\lambda_2} = .288$ and $\sigma_{\beta_1} = \sigma_{\beta_2} = .07$. Given that the time constants T_{Gi}, i = 1, 2, are considerably smaller than the other time constants we have taken as n=5 the state of the system to be identified. Moreover, in view of the fact that the observability indices of the system taken with $T_{Gi} = 0$ are 3 and 2, we have imposed as $n_1 = 3$ and $n_2 = 2$ the unknown observability indices of the system to be identified.

As shown by Figure 5, a typical output of the program gives among other things the entries of the matrices associated with the system operator representation, (1), plus the input-output transfer matrix . For our application example the computed transfer matrice has turned out to be

$$\hat{G}(s) = \frac{1}{\det}\left[\begin{array}{cc} \dfrac{K_1(s+1.24)(s-37.1)}{(s+.47\pm j3.52)} & \dfrac{K_2(s+1.88)(s-6.45)}{(s+5.41\pm j6.00)} \\ \hline \dfrac{K_3(s-2.44)s+.126\pm j10.93)}{(s+34.0\pm j31.4)} & \dfrac{K_4(s+1.36)(s-19.1\pm j31.4)}{(s+.518\pm j3.42)} \end{array}\right]$$

$$\det = (s + 1.61)\ (s + 1.33 \pm j2.42)\ (s + .252 \pm j4.21)$$

A comparison between the responses of the estimated system and of the original system are given in Figures 3 and 4.

CLOSURE

On the basis of the application example discussed in the paper, as well as of a number of other examples treated in [5] and [6] the identification approach proposed by Salut appears to be working quite well. Further work is needed, however, to speed up the computational routines used in the program so as to make a real time identification possible. One step in this direction is the adoption of the numerical approach proposed by Delfour [7]; this should allow us to drastically reduce the time required to solve the Riccati equation.

As to be expected, the examples considered illustrate the importance of using an input "sufficiently rich" in spectral components. Too, it is extremely important to use initial conditions on the augmented state and on the covariance matrix which are not in contradiction with the actual values of the parameters to be identified. In regard to the sensitivity of the algorithm to a more or less precise guessing in the structural parameters n and n_i , an overestimation of their values does not appear to deteriorate the performance of the algorithm. Such an overestimation can be easily corrected as it usually leads to zero pole cancellations. An underestimation of these parameters is on the contrary more difficult to detect and may leed to an unsatisfactory identification. On the other hand an intentional underestimation can offer the advantage to give a sufficiently precise reduced order model. The potential of the algorithm in this direction appears to be quite promising.

REFERENCES

1. Salut, G., Identification optimale des systèmes linéaires stochastiques, thèse de doctorat d'état, Université Paul Sabatier de Toulouse, june 1976.

2. O.I. Elgerd, Electric Energy Systems Theory: An Introduction, Mc Graw-Hill, 1971.

3. Lefebvre, S., Identification des systèmes multivariables à l'aide de la théorie du filtrage et du concept de formes canoniques: Algorithme, Tech. Rep. of Ecole Polytechnique of Montreal, (under preparation).

4. Kalman, R.E., Bucy, R.S., New Results in linear Prediction and Filtering Theory, Transactions of the A.S.M.E., Jnl. of Basic Engineering, pp. 95-108, March 1961.

5. S. Lefebvre, Identification des systèmes multivariables à l'aide de la théorie du filtrage et du concept de formes canoniques: Application, Tech. Rep. of Ecole Polytechnique of Montreal, EP 78-R-31, Septembre 1978.

6. Salut, G., Favier, G., Aguilar-Martin, J., Tracking the Position of an Aircraft from Noisy Measurements ..., International Symposium on Information Theory, June 1976.

7. Delfour, M., Numerical Solution of the Operator Riccati Equation for the Filtering of Linear Stochastic Hereditary Differential Systems, CRM, Montreal, October 1975.

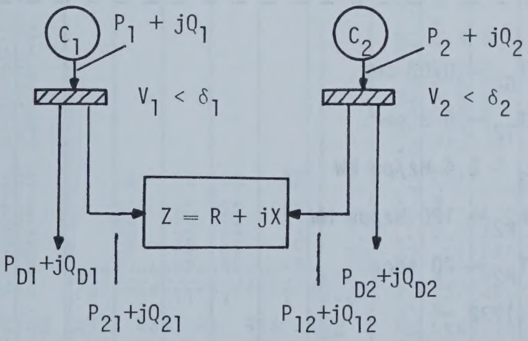
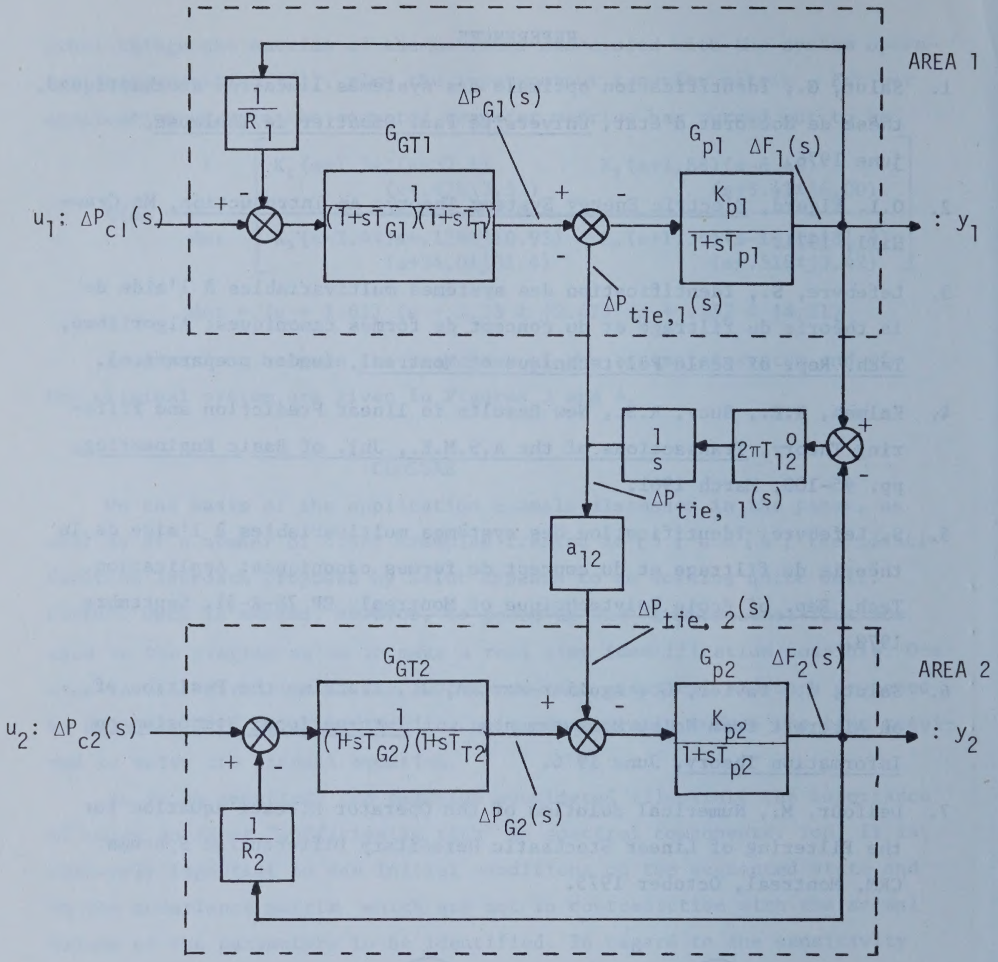


FIGURE 2: A two area interconnected power system.


AREA 1
$\frac{1}{R_1}$
$\Delta P_{G1}(s)$
G_{GT1}
G_{p1}
$\Delta F_1(s)$
$u_1: \Delta P_{c1}(s)$
+
-
$\frac{1}{(1+sT_{G1})(1+sT_{T1})}$
+
-
$\frac{K_{p1}}{1+sT_{p1}}$
$: y_1$
-
$\Delta P_{tie,1}(s)$
$\frac{1}{s}$
$2\pi T^o_{12}$
+
-
$\Delta P_{tie,1}(s)$
a_{12}
$\Delta P_{tie,2}(s)$
G_{GT2}
G_{p2}
$\Delta F_2(s)$
AREA 2
$u_2: \Delta P_{c2}(s)$
+
-
$\frac{1}{(1+sT_{G2})(1+sT_{T2})}$
$\frac{K_{p2}}{1+sT_{p2}}$
$: y_2$
+
-
$\Delta P_{G2}(s)$
$\frac{1}{R_2}$

$T_{G1} = T_{G2} = 0.08$ sec.

$T_{T1} = T_{T2} = 0.3$ sec.

$R_1 = R_2 = 2.4$ Hz/pu MW

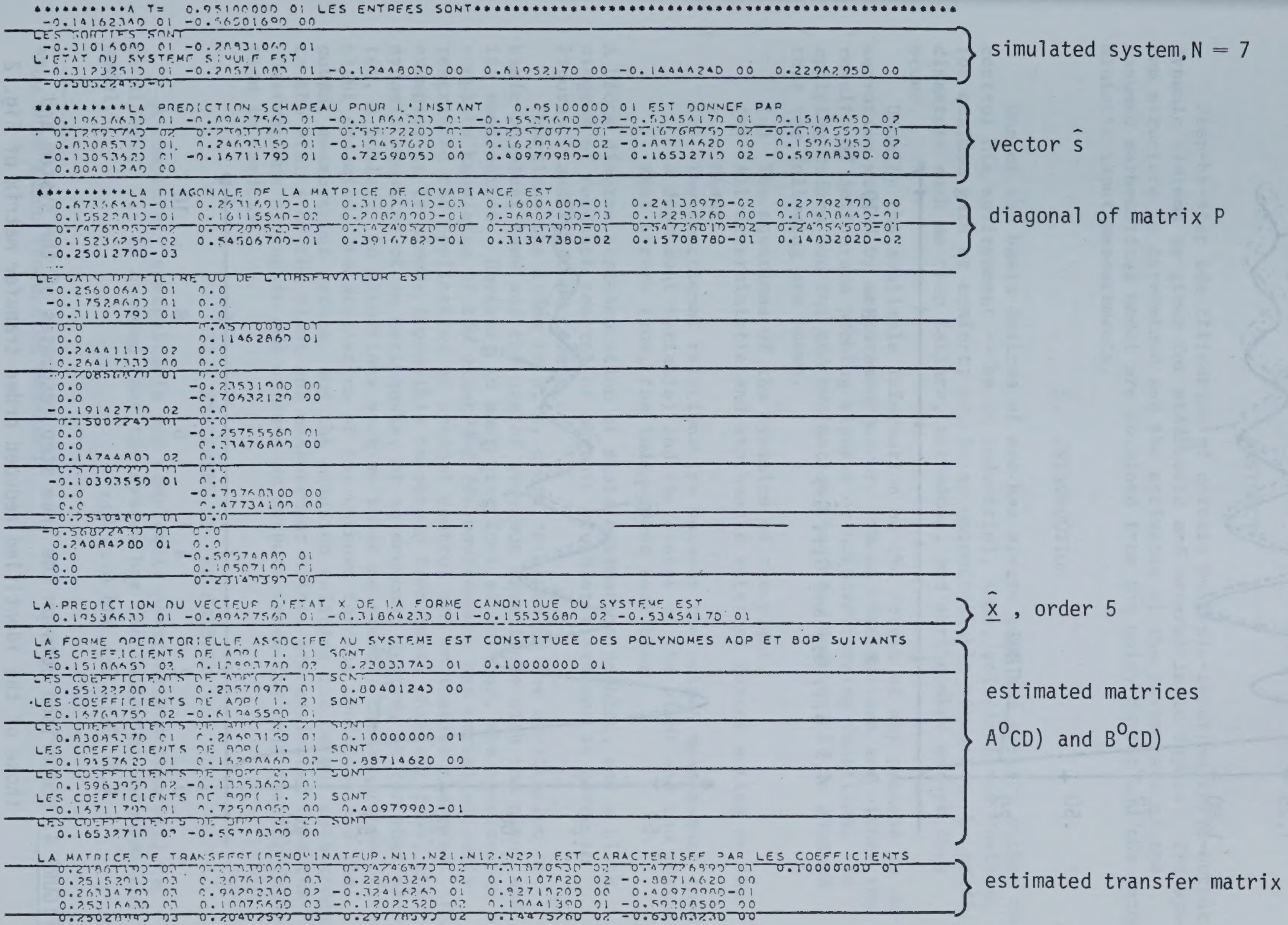
$K_{P1} = K_{P2} = 120$ Hz/pu MW

$T_{P1} = T_{P2} = 20$ sec.

$T_{12} = .1732$

$a_{12} = -1$

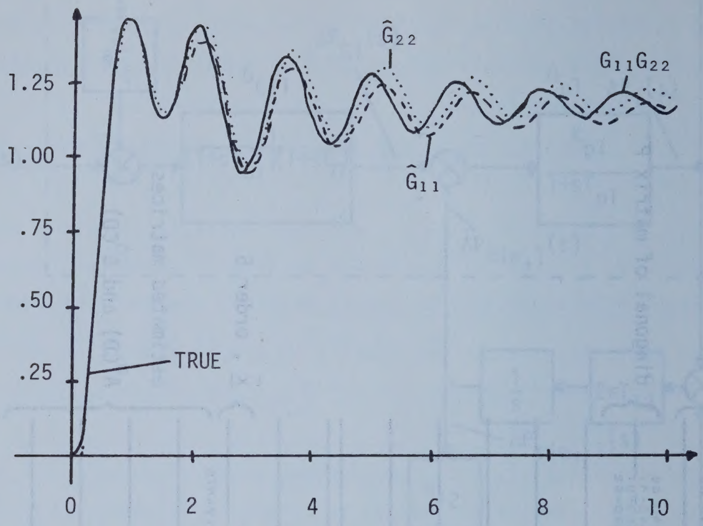
FIGURE 1: The two inputs-two outputs system to be identified.



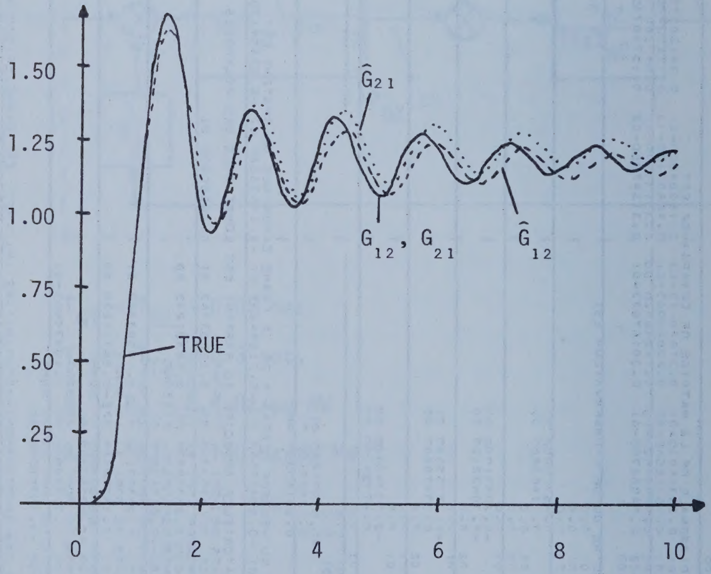
```
************A T=  0.95100000D 01 LES ENTREES SONT*****************************************************
 -0.14162340D 01 -0.56501690D 00
LES SORTIES SONT
 -0.31016080D 01 -0.20931060D 01
L'ETAT DU SYSTEME SIMULE EST
 -0.31232510D 01 -0.20571080D 01 -0.12448030D 00  0.61952170D 00 -0.14444240D 00  0.22962950D 00
 -0.50522450D-01

**********LA PREDICTION SCHAPEAU POUR L'INSTANT    0.95100000D 01 EST DONNEE PAR
  0.19636630D 01 -0.89427560D 01 -0.31864230D 01 -0.15535680D 02 -0.53454170D 01  0.15186650D 02
  0.12893740D 02  0.27031740D 01  0.55122200D 01  0.23570970D 01 -0.16768750D 02 -0.61245590D 01
  0.83085370D 01  0.24693150D 01 -0.19457620D 01  0.16299460D 02 -0.88714620D 00  0.15963950D 02
 -0.13053620D 01 -0.16711790D 01  0.72598950D 00  0.40979980D-01  0.16532710D 02 -0.59788390D 00
  0.80401240D 00

**********LA DIAGONALE DE LA MATRICE DE COVARIANCE EST
  0.67356440D-01  0.26316910D-01  0.31020110D-03  0.16004800D-01  0.24130970D-02  0.22792790D 00
  0.15522810D-01  0.16115540D-02  0.20820920D-01  0.26802130D-03  0.12283260D 00  0.10438440D-01
  0.70768950D-02  0.97720952D-03  0.14240520D 00  0.33131190D-01  0.54736010D-02  0.24056500D-01
  0.15236250D-02  0.54506700D-01  0.39167820D-01  0.31347380D-02  0.15708780D-01  0.14832020D-02
 -0.25012700D-03

LE GAIN DU FILTRE OU DE L'OBSERVATEUR EST
 -0.25600640D 01   0.0
 -0.17528600D 01   0.0
  0.31100790D 01   0.0
  0.0              0.45710000D 01
  0.0              0.11462860D 01
  0.24441110D 02   0.0
  0.26417330D 00   0.0
 -0.70850370D 01   0.0
  0.0             -0.23531900D 00
  0.0             -0.70632120D 00
 -0.19142710D 02   0.0
  0.15007240D 01   0.0
  0.0             -0.25755560D 01
  0.0              0.33476840D 00
  0.14744800D 02   0.0
  0.57107770D 01   0.0
 -0.10393550D 01   0.0
  0.0             -0.73760300D 00
  0.0              0.47734100D 00
 -0.75409800D 01   0.0
 -0.56822430D 01   0.0
  0.24084200D 01   0.0
  0.0             -0.59574880D 01
  0.0              0.10507190D 01
  0.0              0.23140390D 00

LA PREDICTION DU VECTEUR D'ETAT X DE LA FORME CANONIQUE DU SYSTEME EST
  0.19536630D 01 -0.89427560D 01 -0.31864230D 01 -0.15535680D 02 -0.53454170D 01

LA FORME OPERATORIELLE ASSOCIEE AU SYSTEME EST CONSTITUEE DES POLYNOMES AOP ET BOP SUIVANTS
LES COEFFICIENTS DE AOP( 1. 1) SONT
 -0.15186650D 02  0.12893740D 02  0.23033740D 01  0.10000000D 01
LES COEFFICIENTS DE AOP( 2. 1) SONT
  0.55122200D 01  0.23570970D 01  0.80401240D 00
LES COEFFICIENTS DE AOP( 1. 2) SONT
 -0.16769750D 02 -0.61245590D 01
LES COEFFICIENTS DE AOP( 2. 2) SONT
  0.83085370D 01  0.24693150D 01  0.10000000D 01
LES COEFFICIENTS DE BOP( 1. 1) SONT
 -0.19457620D 01  0.16299460D 02 -0.88714620D 00
LES COEFFICIENTS DE BOP( 2. 1) SONT
  0.15963950D 02 -0.13053620D 01
LES COEFFICIENTS DE BOP( 1. 2) SONT
 -0.16711790D 01  0.72598950D 00  0.40979980D-01
LES COEFFICIENTS DE BOP( 2. 2) SONT
  0.16532710D 02 -0.59788390D 00

LA MATRICE DE TRANSFERT(DENOMINATEUR.N11.N21.N12.N22) EST CARACTERISEE PAR LES COEFFICIENTS
  0.21861190D 03  0.21130000D 03  0.94246870D 02  0.31870530D 02  0.47726890D 01  0.10000000D 01
  0.25152910D 03  0.20761200D 03  0.22843240D 02  0.14107820D 02 -0.88714620D 00
  0.26334790D 03  0.94292340D 02 -0.32416260D 01  0.92719200D 00  0.40979980D-01
  0.25216430D 03  0.10075650D 03 -0.12022520D 02  0.19441390D 01 -0.59208500D 00
  0.25020940D 03  0.20407590D 03  0.29778590D 02  0.14475260D 02 -0.63083230D 00
```

simulated system, N = 7
vector $\hat{s}$
diagonal of matrix P
$\underline{\hat{x}}$, order 5
estimated matrices
A^{o}CD) and B^{o}CD)
estimated transfer matrix

FIGURE 3: Estimated values for the system of Figure 1 at 9.5 seconds.



4.1 G_{11}, G_{22} and $\hat{G}_{11}$, $\hat{G}_{22}$



4.2 G_{12}, G_{21} and $\hat{G}_{12}$, $\hat{G}_{21}$

FIGURE 4: Comparison between the true step responses of the system and those of the identified reduced order transfer matrix of Fig. 2

A STEP-BY-STEP PROCEDURE FOR COMPLETE IDENTIFICATION OF CERTAIN MULTIPLE-INPUT/MULTIPLE-OUTPUT STOCHASTIC SYSTEMS

E. D. EYMAN AND E. A. ADEDOTUN*
Information Engineering, University of Iowa, Iowa City, Iowa 52242

ABSTRACT

Step-by-step identification of certain multiple-input/multiple-output dynamic systems is given for stochastic and deterministic inputs. The system structure is determined and the estimates of the parameters of the assumed mathematical model are obtained from the noisy output and the deterministic input measurements.

I. INTRODUCTION

One of the basic desires of man has always been the ability either to control his environment -- be it industrial, home, political, or business -- for maximum gain or comfort; or, in an uncontrollable situation, to predict disasters such as crop failure, tornadoes, and earthquakes before they occur.

Usually the available information on the origin of any process is observation masked by measurement error. These observations are either the results of deliberate probing signals or unknown forcing functions. In applying modern control and estimation theories, however, it is assumed that the following are known.

(a) The equations of the dynamics of the plant.
(b) All deterministic and stochastic external forces acting on the plant.
(c) The functional relationships between the output measurement (now, a dependent variable) and the states of the plant and the measurement errors (now, the independent variables).

A block diagram representation of state estimation problem, and a block diagram of feedback control of certain stochastic systems is given in Figures 1 and 2, respectively.

Sometimes the items in a, b, c are obtained by the application of basic physical laws and the use of previous experience with the plant and its environment. However, in many practical situations, the requirement of complete knowledge of the plant and its environment has unrealistically restricted the application of optimal control and estimation theories. For example, it is almost impossible to obtain from basic physical laws, the dynamic equation of an earthquake, or an economic system, or for that matter, a boiler. In situations such as those mentioned, the equations of the plant dynamics, the statistics of the unknown forcing functions and the output measurement errors, and the equation relating the output measurement to the states of the plant and measurement noise, are estimated from the observations of inputs and outputs or outputs (when the only forcing function is random).

II. STATEMENT OF THE PROBLEM

A completely controllable and completely observable system will be considered. It is assumed that the system has the following mathematical model.

---

*Now with Exxon Production Research

$$\begin{aligned} &\underline{x}(k + L) = A_1\underline{x}(k) + G_1\underline{u}(k) + \underline{e}(k) \\ &\underline{y}(k) = H_1\underline{x}(k) + \underline{\nu}(k) \end{aligned} \tag{1}$$

where $\underline{x}(k) \in R^n$, $\underline{u}(k) \in R^n$, $\underline{y}(k) \in R^m$, $\underline{\nu}(k) \in R^n$, $H_1 \in R^{m\times n}$, $G_1 \in R^{n\times r}$,

$\underline{e}(k) \in R^n$, $E[\underline{e}(k)e'(\ell)] = Q_1\delta_{k\ell}$, $E[\underline{e}(k)\underline{\nu}'(\ell)] = 0$, $E[\underline{\nu}(k)\underline{\nu}'(\ell)] = R_1\delta_{k\ell}$, $E[\underline{x}(0)] = 0$, $E[\underline{x}(0)\underline{x}'(0)] = \gamma$ and $\underline{x}(0)$ is uncorrelated with $\underline{e}(k)$ and $\underline{\nu}(k)$ for all k. $\{\underline{y}(k)\}$ and $\{\underline{u}(k)\}$ are given. For a complete identification n is determined and the estimates of A_1, G_1, H_1, Q_1 γ, R_1 are obtained from input-output measurements.

Now an overview of what is done will be presented in order to satisfy the identifiability conditions; from Tse and Weinart [1], the innovation representation of equation (1) is found. The invariant canonical form of the innovation representation is given by

$$\begin{aligned} &\hat{\underline{x}}(k + 1|k) + A\hat{\underline{x}}(k|k - 1) + G\underline{u}(k) + K\underline{\eta}(k) \\ &\underline{y}(k) = H\hat{\underline{x}}(k|k - 1) + \underline{\eta}(k) \end{aligned} \tag{2}$$

where $\{\eta(k)\}$ is the innovation process. The covariance matrix of $\{\eta(k)\}$ is Q. The parameters to be identified are A, G, K, H, W and n.

Because the assumed model is linear, we have

$$\underline{y}(k) = y_1(k) + y_2(k) \tag{3}$$

where
$$\begin{aligned} &\hat{\underline{x}}_1(k + 1|k) = A\hat{\underline{x}}_1(k|k - 1) + K\underline{\eta}(k) \\ &\underline{y}_1(k) = H\hat{\underline{x}}_1(k|k - 1) + \underline{\eta}(k)\ , \end{aligned} \tag{4}$$

and
$$\begin{aligned} &\underline{x}_2(k + 1) = A\underline{x}_2(k) + G\underline{u}(k) \\ &\underline{y}_2(k) = H\underline{x}_2(k)\ . \end{aligned} \tag{5}$$

Algebraic manipulation yields

$$\begin{aligned} \underline{y}(k) &\approx H \sum_{i=0}^{\ell_2} A^i\ G\underline{u}(k - i - 1) + \alpha(k) - \sum_{i=1}^{\ell_1} A_i\ \underline{y}_1(k - i) \\ &= \sum_{i=0}^{\ell_2} \theta_i\ \underline{u}(k - i - 1) - \sum_{i=1}^{\ell_1} A_i\ \underline{y}_1(k - i) + \underline{\alpha}(k)\ . \end{aligned} \tag{6}$$

where $\theta_i = HA^iG$.

Equation (6) is a regression of $\underline{y}(k)$ on $\underline{u}(k)$ and the residual errors are correlation. ℓ_1 and ℓ_2 are obtained by using $\{\underline{u}(k)\}$ and $\{\underline{y}(k)\}$.

From equation (6), the deterministic part of the output $(\underline{y}(k))$ is $\sum_{i=0}^{\ell_2} \theta_i\ \underline{u}(k - i - 1)$. It is obtained by performing two stages of the method the least squares. In the first stage, the residuals are assumed uncorrelated, the regression parameters are estimated. In the second stage, the estimated parameters are modified to correct for correlation in the residuals. Subtracting the deterministic part from the output, we obtain the stochastic part. Thus, we have separated the output into two parts.

Having obtained $\{\underline{y}_1(k)\}$ and $\{\underline{y}_2(k)\}$, the parameters of equations (4) and (5) can be estimated. If the parameters of equation (4) are estimated first, the parameters of equation (5) are completely determined by estimating G. Also, if the parameters of equation (5) are estimated first, the parameters of equation (4) are completely determined by estimating K and the covariance of $\{\underline{\eta}(k)\}$. In this paper equation (4) is used first. Therefore, after we have completely identified (4), we estimate the gain matrix G of (5).

III. ALGORITHMS FOR THE SOLUTION OF THE PROBLEM

From equation (3), $\underline{y}(k) = \underline{y}_1(k) + \underline{y}_2(k)$. (7)

From equation (5), $\underline{y}_2(k) = \sum_{i=0}^{\ell_1} \theta_i \underline{u}(k - i - 1)$ (8)

Also, $\underline{y}_1(k) = -\sum_{i=1}^{\ell_2} \beta_i \underline{y}_1(k - i) + \underline{\alpha}(k)$. (9)

Assume N_1 data points are available, where N_1 is greater than or equal to $N + j$ where j is greater than either ℓ_1 or ℓ_2. The following definitions will be used in the regression equations that follow.

$$y \triangleq \begin{vmatrix} \underline{y}^T(j) \\ \underline{y}^T(j+1) \\ \vdots \\ \underline{y}^T(j+N) \end{vmatrix} \qquad y_1 \triangleq \begin{vmatrix} \underline{y}_1^T(j) \\ \underline{y}_1^T(j+1) \\ \vdots \\ \underline{y}_1^T(j+N) \end{vmatrix}$$

$$\psi = \begin{vmatrix} \psi_j \\ \psi_{j+1} \\ \vdots \\ \psi_{j+N} \end{vmatrix} \qquad \theta = \begin{vmatrix} \theta_1^T \\ \theta_2^T \\ \vdots \\ \theta_{\ell_1}^T \end{vmatrix}$$

$$\psi_i \triangleq [\underline{u}^T(j - 1) \cdots \underline{u}^T(j - \ell_1 - 1)] .$$

$$\beta \triangleq \begin{vmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_{\ell_2}^T \end{vmatrix} \quad \phi \triangleq \begin{vmatrix} \phi_j \\ \phi_{j+1} \\ \vdots \\ \phi_{j+N} \end{vmatrix} \quad \alpha \triangleq \begin{vmatrix} \underline{\alpha}^T(j) \\ \underline{\alpha}^T(j+1) \\ \vdots \\ \underline{\alpha}^T(j+N) \end{vmatrix}$$

$$\phi \triangleq [-\underline{y}_1^T(j - 1), -\underline{y}_1^T(j - 2), \ldots, -y_1^T(j - \ell_2)] .$$

Thus, $y = \psi\theta + y_1$ and $y_1 = \phi\beta + \alpha$. (10)

From equation (10), $y = \psi\theta + \phi\beta + \alpha = [\psi \quad \psi] \begin{matrix}\theta \\ \beta\end{matrix} + \alpha$. (11)

(ℓ_1 and ℓ_2 are still unknown.) To determine ℓ_1, use equation (10) and assume, for now, that $\{\underline{y}_1(k)\}$ is white noise

$$y = \psi\theta + y_1 \; . \tag{12}$$

Starting with $\ell_1 = 0$, use the following steps to obtain ℓ_1^*.

(a) Find $\hat{\theta}$ from (12); $\hat{\theta} = (\psi^T\psi)^{-1} \psi^T y$. (13)

(b) Obtain $\hat{y}_1$ from equations (12) and (13); $\hat{y}_1 = y - \psi(\psi^T\psi)^{-1} \psi^T y$.

(c) Obtain $\det \frac{1}{N} \hat{y}_1^T \hat{y}_1$.

(d) Increase ℓ_1 by 1 and go back to a.

(e) Let ℓ_1^* be ℓ_1 for the minimum in c, and $\hat{y}_1^*$ be $\hat{y}_1$ obtained in (b) when ℓ_1 is ℓ_1^*. By definition $\{\underline{y}_1^*(k)\}$ is now available.

$$\underline{y}_1^*(k) = - \sum_{i=1}^{\ell_2} \beta_2 \underline{y}_1^*(k - i) + \underline{\alpha}(k) \; . \tag{14}$$

Because $\{\underline{y}_1^*(k)\}$ is available, a one-step prediction method as given by Jones 2 is used to obtain ℓ_2^* where ℓ_2^* is the order that gives the minimum one-step prediction error.

Our next step is the estimation of θ. From equation (11)

$$\alpha = y - [\psi \quad \theta] \begin{bmatrix}\theta \\ \beta\end{bmatrix} \; . \tag{15}$$

Let $J \overset{\Delta}{=} \text{Trace} \frac{1}{N} \alpha^T \alpha$, where J is the objective function to be minimized. The least squares estimates of $\hat{\theta}$ and $\hat{\beta}$ are obtained from

$$\begin{vmatrix}\hat{\theta} \\ \hat{\beta}\end{vmatrix} = \begin{vmatrix}\psi^T\psi & \psi^T\phi \\ \phi^T\psi & \phi^T\phi\end{vmatrix} \begin{vmatrix}\psi^T y \\ \phi^T y\end{vmatrix} .$$

Using the partitioned matrix

$$\hat{\theta} = [\psi^T\psi]^{-1} \psi^T y - [\psi^T\psi]^{-1} \psi^T\phi F^{-1} \phi^T M y \; , \quad \hat{\beta} = F^{-1} \theta^T M y$$

where $M = I = \psi(\psi^T\psi)^{-1} \psi^T$ and $F = \phi^T M \phi$.

$$\hat{\theta} = (\psi^T\psi)^{-1} \psi^T y - (\psi^T\psi)^{-1} \psi^T\phi\hat{\beta} \tag{16}$$

$$\overset{\Delta}{=} \hat{\theta}_{LS} - \hat{\theta}_{BIAS} \; .$$

We use the following algorithm to minimize $\hat{\theta}_{BIAS}$. The algorithm is a modification of that given by Hsia [3].

Step 1. Assume $\hat{\beta} = 0$ in equation (16) and compute $\hat{\theta}_{LS} = (\psi^T\psi)^{-1} \psi^T y$.

Step 2. Set $\hat{\theta} = \hat{\theta}_{LS}$ and generate $\hat{\underline{y}}_1^T(k) = \underline{y}^T(k) - \psi_k\hat{\theta}$. Compute

$$\hat{F} = \hat{\phi}^T M \hat{\phi} \; .$$

Step 3. Compute

$$\hat{\beta} = \hat{F}^{-1} \phi^T My \; ; \quad \theta_{BIAS} = (\psi^T\psi)^{-1} \psi^T \phi \hat{G} \; .$$

Update $\hat{\theta}$: $\quad \hat{\theta} = \hat{\theta}_{LS} - \hat{\theta}_{BIAS}$.

Step 4. Continue from Step 2 until the norm of $\hat{\theta}_{BIAS}$ is as small as desired. With $\hat{\theta}$ so obtained we find $\{\underline{y}_1(k)\}$ by using $\hat{\underline{y}}_1^T(k) = \underline{y}^T(k) - \psi_k\hat{\theta}$. The output measurements are now separated into deterministic and stochastic parts:

$$\hat{\underline{y}}_1^T(k) = \underline{y}^T(k) - \psi_k\hat{\theta} \; ; \quad \hat{\underline{y}}_2^T(k) = \psi_k\hat{\theta} \; . \tag{17}$$

From the set of equations (4)

$$\hat{\underline{x}}_1(k + 1|k) = A\hat{\underline{x}}_1(k|k - 1) + K\underline{\eta}(k) \; ; \quad \hat{\underline{y}}_1(k) = H\hat{\underline{x}}_1(k|k - 1) + \underline{\eta}(k) \; .$$

Using the method of Tse and Weinert [1] the order n and the parameters A, H, K and the covariance of $\{\underline{\eta}(k)\}$ are obtained [5]. Substituting the the estimates of A and H into equation (5) we have

$$\underline{x}_2(k + 1) = \hat{A}\underline{x}_2(k) + G\underline{u}(k) \; ; \quad \hat{\underline{y}}_2(k) = \hat{H}x_2(x) \; . \tag{18}$$

From (8) we have $\hat{\underline{y}}_2(k) = \hat{H}[zI - \hat{A}]^{-1} \; G\underline{u}(k)$.

$$|zI - \hat{A}| \; \hat{\underline{y}}_2(k) = \hat{H} \; \mathrm{Adj}[zI - \hat{A}]Gu(k) \; . \tag{19}$$

From equation (17), $\hat{\underline{y}}_2(k) = \hat{\theta}^T \; \psi_k^T$. Substitute into (19) to obtain

$|zI - \hat{A}|\hat{\theta}^T \; \psi_k^T = \hat{H} \; \mathrm{Adj}[zI - \hat{A}]G\underline{u}(k)$. This equation can now be solved to obtain G.

3.1. Sample Problem for the Step-by-Step Procedure

In order to demonstrate the step-by-step procedure given in this paper, the following three sample problems will be considered.

(1) A single-input/single-output first order system.
(2) A single-input/two-output second order system.
(3) A single-input/two-output third order system.

In each case both measurement and system noise are present. Simulated systems are used so that the results of the step-by-step identification procedure could be compared with known values.

In order to avoid repetition in describing the steps in identifying each of the sample problems, the sequence of calculations will now be discussed in detail. For this detailed description, we use a general set of dynamic and measurement equations for a linear discrete, time-invariant mathematical model. Consider the following set of equations.

$$\underline{x}(k + 1) = A\underline{x}(k) + G\underline{u}(k) + c\underline{e}(k)$$

$$\underline{y}(k) \; H\underline{x}(k) + \underline{\nu}(k)$$

where the pair (A,H) is in an invariant canonical form.

$E[\underline{e}(k)] = 0$, $\quad E[\underline{\nu}(k)] = 0$, $\quad E[\underline{e}(k)\underline{e}^T(\ell)] = I\delta_{k\ell}$ and

$E[\underline{\nu}(k)\underline{\nu}T(\ell)] = (0.1)I\delta_{k\ell}$.

Assume that the system is a combination of a stochastic and a deterministic subsystem. Also, assume that the mathematical models of the subsystems have the following sets of equations:

$$\underline{x}_1(k + 1) = A\underline{x}_1(k) + c\underline{e}(k) \; ; \quad \underline{y}_1(k) = H\underline{x}_1(k) + \underline{\nu}(k) \tag{20}$$

$$\underline{x}_2(k + 1) = A\underline{x}_2(k) + G\underline{u}(k) \; ; \quad \underline{y}_2(k) = H\underline{x}_2(k) + \underline{\nu}(k) \tag{21}$$

and $\underline{y}(k) = \underline{y}_1(k) + \underline{y}_2(k)$.

From the sets of equations (20) and (21) it is seen that A and H are common to both sets. Therefore, A and H can be estimated by using either of the two sets. In this paper the set of equations (20) is used

The steps in the identification procedure will now be described. Figures 4 and 5 are block diagram representations of the method. Assume that the stochastic input signal is optimal for identification -- the stochastic optimal inputs for the sample problems are obtained by changing the values of the elements of matrix C until optimal values for identification are obtained. The optimal values are those that give the smallest error in estimating the elements of the canonical matrices A and H. A step-by-step procedure for obtaining the optimal matrix C is as described in Step 0.

Step 0

a. Choose a set of values for the elements of matrix C.

b. Substitute these values into the set of equations (20). (Note that A, H, u(k), G, and the covariance matrices of $\{\underline{e}(k)\}$ and $\{\underline{\nu}(k)\}$ are given.) Generate and store an independent sequence of random numbers for the random sequences $\{\underline{e}(k)\}$ and $\{\underline{\nu}(k)\}$. Using the set of equations (20) generate and store the sequences $\{\underline{y}(k)\}$ and $\{\underline{u}(k)\}$.

c. In order to obtain the deterministic sequence $\{\underline{y}_2(k)\}$ of equation (21) from the output sequence $\{\underline{y}(k)\}$ of equation (20), regress $\underline{y}(k)$ on $\underline{u}(k)$ as follows:

   (i) Choose the highest order to be tested in the regression that follows.

   (ii) Calculate the determinant of the covariance matrix of the residuals of regression using all the orders from zero to the highest order chosen. Choose, as the order of regression, the order that gives the minimum value of the determinants calculated. Using the regression parameters, calculate the deterministic part $\{\underline{y}_2(k)\}$ of the output sequence $\{\underline{y}(k)\}$.

   (iii) Subtract the deterministic sequence $\{\underline{y}_2(k)\}$ from the output sequence $\{\underline{y}(k)\}$ to obtain the sequence of residuals $\{\underline{y}_1(k)\}$. Assume that the residuals form an autoregressive process and obtain the order of the process by using the Akaike's Information Criterion.

   (iv) Because the orders of the regression and the autoregressive process are available, obtain the estimates of the regression parameters as follows. First assume that the residuals are uncorrelated; then regress $\underline{y}(k)$ on $\underline{u}(k)$; finally, modify the estimates to correct for errors because the residuals are correlated. Using these estimates of the regression parameters calculate the deterministic sequence of the output sequence.

   (v) Subtract the deterministic sequence $\{\underline{y}_2(k)\}$ obtained in (iv) from the output sequence. The residual sequence is the stochastic part, $\{\underline{y}_1(k)\}$, of the output sequence.

(vi) Choose the value of the highest lag big enough so that the covariance matrices of the residuals calculated are sufficient for the structural indices determination and the parameter estimation.

(vii) Calculate the covariance matrices from lag one through the highest lag chosen.

(viii) Calculate the determinants that will be used in the structural determination and parameter estimation.

(ix) From the plots of the determinants, obtain the structural indices.

(x) Calculate the estimates of A and H using the method in Appendix A.

(xi) If the estimates of A and H have peaked, go to step one; otherwise change some element of C and go back to (a).

In the next six steps it is assumed that either the stochastic input is optimal or the optimal gain matrix C of the stochastic input has been obtained in Step 0.

Step 1. Substitute the optimal matrix C into the set of equations (20). Generate and store the sequences $\{\underline{y}(k)\}$ and $\{\underline{u}(k)\}$.

Step 2. Repeat (i) through (iv) of (c) to Step 0.

Step 3. Repeat (iv) through (ix) of (c) to Step 0.

Step 4. Repeat (x) of (c) of Step 0.

Step 5. Having obtained the final estimates of A and H in Step 1, substituting these estimates of A and H in the set of equations (20), we have

$$\underline{x}_1(k+1) = \hat{A}\underline{x}_1(k) + c\underline{e}(k) \; ; \quad \hat{\underline{y}}_1(k) = \hat{H}x_1(k) + \underline{\nu}(k) \; . \tag{22}$$

The set of equations (22) has an innovation representation:

$$\begin{aligned} \hat{\underline{x}}_1(k+1|k) &= \hat{A}\hat{\underline{x}}(k|k-1) + K\underline{\eta}(k) \\ \hat{\underline{y}}_1(k) &= \hat{H}\hat{\underline{x}}(k|k-1) + \underline{\eta}(k) \; . \end{aligned} \tag{23}$$

Calculate, using the iterative process described in Appendix A, the estimates of K and Q (the covariance matrix of the innovations sequence $\{\underline{\eta}(k)\}$). Obtain the actual values of K and Q by using the Kalman-Bucy equations.

Step 6. In the last step the deterministic gain matrix is estimated. This is done by substituting the estimates $\hat{A}$ and $\hat{H}$ for A and H in equation (21). Then, equation (21) becomes

$$\underline{x}_2(k+1) = \hat{A}\underline{x}_2(k) + G\underline{u}(k) \; ; \quad \hat{\underline{y}}_2(k) = \hat{H}\underline{x}_2(k) \; . \tag{24}$$

By finding the transfer function of (24), we obtain an expression for $\hat{\underline{y}}_2(k)$ in terms of $\underline{u}(k)$. That is

$$\hat{\underline{y}}_2(k) = \hat{H}[I_z - \hat{A}]^{-1} G\underline{u}(k) \tag{25}$$

Also, in Step 2 we obtain the deterministic part $\{\underline{y}_2(k)\}$ of the output $\{\underline{y}(k)\}$.

Set the value of $\hat{\underline{y}}(k)$ in Step 2 equal to the right-hand side of equation (25). Calculate G.

IV. SAMPLE PROBLEMS

The problems considered were (1) a single-input/single-output first order system, (2) a single-input/two-output second order system, and (3) a single-input/two-output third order system. Part (1) will be shown here to illustrate the method.

4.1. Single-Input/Single-Output First Order System

Consider the system's dynamic and measurement equations:

$$x(k+1) = 0.7\underline{x}(k) + 1.0u(k) + Ce(k); \quad y(k) = x(k) + \nu(k) \tag{26}$$

where $E[e(k)] = 0$, $E[e(k)e(\ell)] = 1\delta_{k\ell}$, $u(k) = 30.0$, $E[\nu(k)] = 0$, $E[\nu(k)\nu(\ell)] = 0.1\delta_{k\ell}$.

Step 0. $C = 1.0$.

Step 1. The dynamic and the measurement equations

$$x(k+1) = 0.7x(k) + u(k) + e(k); \quad y(k) = x(k) + \nu(k) \tag{27}$$

are used to generate the input-output sequences $\{u(k)\}$, $\{y(k)\}$. The number of points generated and stored is 3000.

Step 2. With the date in Step 1 the maximum regression order of 4 is used in the separator program. The number of data points used is 450. The computer output is in Table 1. The order selected is 1.

TABLE 1. SELECTION OF REGRESSION ORDER FOR SAMPLE PROBLEM 1

Order	Residuals		
	Mean	Standard Deviation	Determinant of Covariance
0	0.0039	1.92	3.69
1	-0.0029	1.91	3.65
2	-0.35	1.91	3.77
3	-0.26	1.91	3.72
4	-0.106	1.91	4.77

A selection of the order of autoregression of the residual is made by using Akaike's Information Criterion. Order selected is 1. The regression parameters are: 103.48, -100.16. (See Table 2.)

Step 3. The structural index is determined next: 28000 data points. The first big decrease occurs at lag 1; therefore, the dynamic order is 1.

Now, the steady state innovations representation has the form

$$\begin{aligned} \hat{x}(k+1|k) &= \beta_o \hat{x}(k|k-1) + u(k) + K\eta(k) \\ y(k) &= \hat{x}(k|k-1) + \eta(k) \end{aligned} \tag{28}$$

where $E[\eta(k)\eta(\ell)] = q\delta_{k\ell}$.

Step 4. From Table 3, $\beta_o = \dfrac{1.64}{2.64} = 0.699$.

Step 5. K, g and their estimates are computed by using Kalman-Bucy filter equations and the iteration in reference 5. Actual gain $K = 0.667$, actual covariance of innovation $q = 2.5$, estimated gain $\hat{K} = 0.61$, estimated covariance $\hat{q} = 2.09$.

TABLE 2. SELECTION OF ORDER OF AUTOREGRESSION OF RESIDUALS IN SAMPLE PROBLEM 1

Order	AIC	Order	AIC
0	583.48	11	343.27
1	328.15	12	343.92
2	330.00	13	345.92
3	332.00	14	345.92
4	333.95	15	347.79
5	335.92	16	349.68
6	337.92	17	351.43
7	338.38	18	353.20
8	338.84	19	353.96
9	339.67	20	354.29
10	341.66		

TABLE 3. DETERMINATION OF THE STRUCTURAL INDEX FOR SAMPLE PROBLEM

Lag	Residual Covariance
0	3.61
1	2.343
2	1.64
3	0.979
4	0.624
5	0.387
6	0.26
7	0.115
8	0.084

Step 6. From Step 2, the regression lag is 1 and the parameter estimates are: 103.48 and -100.16 which implies

$$\hat{y}_2(k) = 103.48u(k - 1) - 100.16u(k - 2) . \tag{29}$$

Also, replacing the gain of $u(k)$ by g in (27)

$$y_2(k) = \frac{1}{0.7 - z} gu(k) \quad \text{and} \quad y_2(k)(z - 0.7) = gu(k) . \tag{30}$$

Substitute (29) into (30): $[103.48 \times u(k - 1) - 100.16 \times u(k - 2)]$
$-0.7[103.48 \times u(k - 2) - 100.16 \times u(k - 2)] = gu(k)$.

Because $u(k)$ is a constant for all k, $g = (1 - 0.7)3.32 = 0.996$.

TABLE 4. COMPARISON OF ACTUAL AND ESTIMATED VALUES OF SAMPLE PROBLEM 1

Parameter	Actual Value	Estimated Value
β_o	0.7	0.699
g	1.0	0.996
K	0.667	0.61
q	2.5	2.09

4.2. Single-Input/Two-Output Second Order System

$$\underline{x}(k + 1) = \begin{vmatrix} 0.0 & 1.0 \\ 0.2 & 0.3 \end{vmatrix} \underline{x}(k) + \begin{vmatrix} 1.0 \\ 1.0 \end{vmatrix} u(k) + C\underline{e}(k)$$

$$\underline{y}(k) = \begin{vmatrix} 1.0 & 0.0 \\ 1.2 & 1.6 \end{vmatrix} \underline{x}(k) + \underline{\nu}(k)$$

TABLE 5. COMPARISON OF ESTIMATED AND ACTUAL VALUES OF SAMPLE PROBLEM 2

Parameter	Actual	Estimated
β_0	0.2	0.206
β_1	0.3	0.301
α_0	1.2	1.18
α_1	1.6	1.64
K_{11}	-0.57	-0.6015
K_{12}	-0.012	-0.024
K_{21}	0.588	0.598
K_{22}	0.184	0.196
Q_{11}	1.77	1.75
Q_{12}	6.59	6.52
Q_{22}	33.69	33.99
G_{11}	1.0	0.9
G_{12}	1.0	1.0

4.3. Single-Input/Two-Output Third Order System

$$\underline{x}(k+1) = \begin{vmatrix} 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \\ 0.024 & -0.26 & 0.9 \end{vmatrix} \underline{x}(k) + \begin{vmatrix} 1.0 \\ 1.0 \\ 1.0 \end{vmatrix} u(k) + C\underline{e}(k)$$

$$\underline{y}(k) = \begin{vmatrix} 1.0 & 0.0 & 0.0 \\ 2.0 & 3.0 & 4.0 \end{vmatrix} \underline{x}(k) + \underline{\nu}(k)$$

TABLE 6. COMPARISON OF ESTIMATED AND ACTUAL VALUES OF SAMPLE PROBLEM 3

Parameter	Actual	Estimated	Parameter	Actual	Estimated
β_0	0.024	0.026	K_{11}	-0.25	-0.26
β_1	-0.26	-0.34	K_{21}	-0.28	-0.28
β_2	0.9	1.0	K_{31}	-0.168	-0.22
α_0	2.0	2.0	K_{12}	0.047	0.047
α_1	3.0	2.8	K_{22}	0.21	0.22
α_2	4.0	4.1	K_{23}	0.18	0.17
G_{11}	1.0	1.0	Q_{11}	1.74	1.73
G_{21}	1.0	1.0	Q_{12}	0.29	-0.62
G_{31}	1.0	1.3	Q_{22}	8.95	8.01

REFERENCES

1. Tse, E., and Weinert, H.L., "Structure determination and parameter identification for multivariable stochastic linear system," IEEE Trans. on Auto. Control, 1975, pp. 603-613
2. Jones, R.J., "Multivariate maximum entropy spectral analysis," Applied Time Series Analysis, symposium sponsored by Univ. of Tulsa, 1976.
3. Hsia, T.C., "On least squares algorithms for system parameter identification," IEEE Trans. on Auto. Control, 1976, pp. 104-108.
4. Jones, R.H., "Identification and autoregressive spectrum estimation," IEEE Trans. on Auto. Control, 1974, pp. 894-898.
5. Adedotun, E.A., 1977, Ph.D. thesis, Univ. of Iowa, Iowa City, Iowa.

THE EFFECTS OF RANDOM CHANNEL ERRORS ON ADAPTIVE DELTA MODULATION ENCODED SIGNALS

D. P. PREZAS and J. L. LoCICERO
Department of Electrical Engineering
Illinois Institute of Technology
Chicago, Illinois 60616

ABSTRACT

This paper considers the effects of random channel errors when the transmitted signal is in adaptive delta modulation (ADM) format. By utilizing the characteristics of an all-digital ADM, we develope a mathematical model which describes how the decoded signal is effected by channel noise. We show that the net effect of a bit error is a scaling of the ADM estimate signal. Utilizing a probabilistic analysis, we demonstrate the limitations of the ADM system for a number of bit error rates and show how the error analysis dictates possible ways to improve performance in a noisy environment. As a conclusion, we present computer simulation results to verify our analysis with a variety of test signals.

INTRODUCTION

Digital delta modulation is a technique, diagramatically shown in Fig. 1, by which an analog signal, x(t), is encoded into a sequence of binary digits (bits), $\{e_x(k)\}$, by periodically comparing the input signal to an estimate signal, $\hat{x}(t)$. The amount of change that the estimate undergoes is called the DM step size, $s_x(k)$. The most general mathematical description of a digital DM is given by the following set of equations:

$$e_x(k) = \operatorname{sgn}[\xi_x(k)], \tag{1}$$

$$\xi_x(k) = x(k) - \hat{x}(k) \tag{2}$$

and

$$\hat{x}(k) = \hat{x}(k-1) + s_x(k), \tag{3}$$

where k denotes the kth time interval.

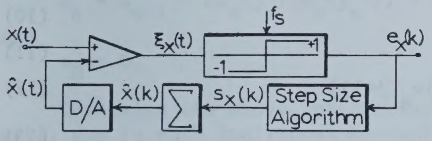


Fig. 1. Basic Digital DM

The particular type of DM is characterized by the step size algorithm. In the ensuing development, we will be concerned with an adaptive step size algorithm used to encode voice-type signals. Thus, we will focus on the Song audio mode ADM [1], where the step size changes linearly, i.e.,

$$s_x(k) = |s_x(k-1)|e_x(k-1) + Se_x(k-2). \tag{4}$$

Here S denotes the magnitude of the minimum step size. For this algorithm it can be shown [2,3] that for $s_x(k-1) \neq 0$,

$$s_x(k) = s_x(k-1)e_x(k-1)e_x(k-2) + Se_x(k-2). \tag{5}$$

Our analysis of the problem of random channel errors ultimately considers the effects on the ADM decoded signal due to a change in the transmitted bit from a 1 to a 0 or vice-versa.

A SINGLE BIT ERROR IN ADM

Let's define the error in an ADM decoded signal, E(k), as the difference between decoder outputs with and without channel errors, i.e.,

$$E(k) = \hat{x}_e(k) - \hat{x}(k). \tag{6}$$

A repetitive application of Eq.(3) results in the following general expression for $\hat{x}(k)$

$$\hat{x}(k) = \hat{x}(-1) + \sum_{m=0}^{k} s_x(m). \tag{7}$$

Utilizing the Song audio mode ADM (Eq.(5)), and assuming $s_x(-1) = 0$, we obtain the following expression for $s_x(m)$,

$$s_x(m) = Se_x(m-2) + Se_x(m-1) \sum_{i=-2}^{m-3} e_x(i)e_x(i+1). \tag{8}$$

Combining Eqs.(7) and (8), we obtain $\hat{x}(k)$ in terms of the transmitted sequence, $\{e_x(k)\}$, i.e.,

$$\hat{x}(k) = \hat{x}(-1) + S \sum_{m=0}^{k} e_x(m-2) + S \sum_{m=1}^{k} e_x(m-1) \sum_{i=-2}^{m-3} e_x(i)e_x(i+1). \tag{9}$$

Let's first calculate $E(k)$ for the case of a single bit error at position $k=j$. The only terms of $\hat{x}(k)$ contributing to $E(k)$ are those containing the error bit $e_x(j)$. When we refer to the error bit $e_x(j)$ we imply its correct value, i.e., before it has been changed due to channel noise. Thus, if a term in $\hat{x}(k)$ appears as $De_x(j)$, where D is independent of $e_x(j)$, then it will appear as $-De_x(j)$ in the expression for $\hat{x}_e(k)$ and the final contribution to $E(k)$ will be $-2De_x(j)$. Applying this to Eq.(9) for error bit $e_x(j)$, we obtain

$$E(k) = 0, \ k \leq j, \tag{10}$$

$$E(j+1) = A_j, \tag{11}$$

$$E(j+2) = E(j+1) - 2Se_x(j)[1+e_x(j-1)e_x(j+1)] \tag{12}$$

and

$$E(k) = E(k-1) + aSe_x(k-1), \ k \geq j+3, \tag{13}$$

where

$$A_j = -2Se_x(j)B_j, \tag{14}$$

$$B_j = \sum_{i=-2}^{j-2} e_x(i)e_x(i+1) \tag{15}$$

and

$$a = -2e_x(j)[e_x(j-1) + e_x(j+1)]. \tag{16}$$

If we solve the difference equation given above (13), we find that

$$E(k) = E(j+2) + aS \sum_{m=j+3}^{k} e_x(m-1), \ k \geq j+3. \tag{17}$$

The first term of Eq.(17) constitutes the D.C. portion of $E(k)$. The nature of audio signals yields a B_j which is always positive. As a result, $E(j+2)$ may be zero or it will have a sign opposite to that of $e_x(j)$. Thus, the D.C. offset due to a single channel error has the polarity of the error bit after the effect of channel noise. The second term of Eq.(17) says that the A.C. portion of $E(k)$ is obtained by linearly decoding the incoming bit stream using a fixed step size equal to aS. We see from Eq.(16) that the coefficient "a" can take on values 0, +4 and -4. When a=0 there is no A.C. distortion. If a=+4 (-4) the output of the linear decoder with a fixed step size 4S is added to (subtracted from) $\hat{x}(k)$ to form $\hat{x}_e(k)$.

MULTIPLE BIT ERRORS IN ADM

Let's consider the case of an error at positions $k = j_1, j_2, \ldots, j_n$. Using the results from the single bit error case and the fact that terms in $\hat{x}(k)$ containing the product of two error bits will not change sign allows us to express the error as

$$E(j_n+1) = E(j_n) + A_{j_n}, \tag{18}$$

$$E(j_n+2) = E(j_n+1) - 2Se_x(j_n)[1+e_x(j_n-1)e_x(j_n+1)] + a_{n-1}Se_x(j_n+1), \tag{19}$$

$$E(k) = E(j_n+2) + a_n S \sum_{m=j_n+3}^{k} e_x(m-1), \quad j_n+3 \leq k \leq j_{n+1} \tag{20}$$

where

$$A_{j_n} = -2Se_x(j_n)B_{j_n}, \tag{21}$$

$$B_{j_n} = B_{j_{n-1}} + \sum_{i=j_{n-1}+1}^{j_n-2} e_x(i)e_x(i+1), \quad n \geq 2, \tag{22}$$

$$B_{j_1} = \sum_{i=-2}^{j_1-2} e_x(i)e_x(i+1), \tag{23}$$

$$a_n = a_{n-1} - 2e_x(j_n)[e_x(j_n-1) + e_x(j_n+1)] \tag{24}$$

and

$$a_0 = 0. \tag{25}$$

Observing the above, particularly Eq.(22), we note that the D.C. offsets, due to each individual channel error, are accumulated. We also note that the fixed step size of the linear decoder, indicated in Eq.(20), whose output represents the A.C. portion of E(k), can take on the values 0 or 4Sr, where r is a positive integer.

POWER SPECTRAL DENSITY OF D.C. NOISE LEVELS

To help understand the effect of channel errors on an ADM encoded signal, we investigate the power spectral density (PSD) of the D.C. portion of the noise. Let n(t), in Fig. 2, represent the D.C. portion of E(k). Each level of n(t) depicts the D.C. offset corresponding to each channel error. The time at which we have a level change corresponds to the random time of occurance of each bit error. We necessarily assume that the random process n(t) is stationary and that the average channel error rate, p, and the average power of the waveform are both constant.

In our effort to calculate the PSD of n(t) we realize that n(t) can be considered as the output of an ideal integrator whose input is the waveform n'(t), also depicted in Fig. 2. It can be shown [4] that the PSD of n'(t) is given as

$$G_{n'}(f) = (pf_s/100)\, \overline{I^2}, \tag{26}$$

where I, the random strength of each impulse in n'(t), represents the difference between two consequtive D.C. levels of n(t). Applying the transfer function of an ideal integrator for $f \neq 0$, the PSD of n(t) can be expressed as

$$G_n(f) = (pf_s/400\pi^2 f^2)\ \overline{I^2},\quad f \neq 0. \tag{27}$$

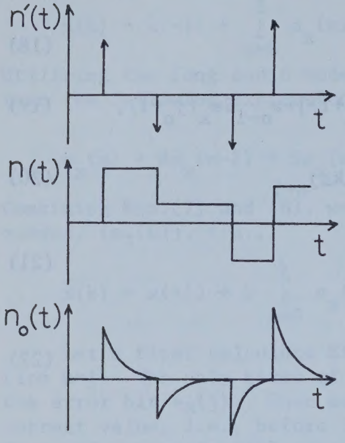
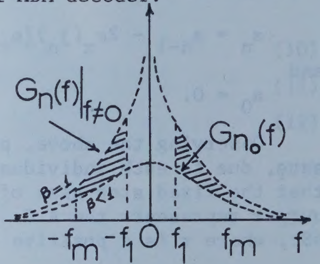


Fig. 2. Waveforms for n'(t), n(t) and $n_0(t)$

Figure 3 depicts the shape of $G_n(f)$. We observe that $G_n(f) \to \infty$ as $f \to 0$, and more importantly that the integral of $G_n(f)$, over a range which includes f=0, is infinite. To minimize the effect of the D.C. levels of E(k) we must use a band pass filter following the decoder. The filter should not, of course, pass D.C. and in addition it should have a lower cut-off frequency, f_1, as large as possible.

The practical design of such a band pass filter will result in some imperfections. As a consequence, the magnitude of the filter function in the stop band around f=0 may not be absolutely zero. Due to the shape of $G_n(f)$ and the filter imperfections, the spectrum of the filtered noise will still represent a fair amount of noise between D.C. and the lower cut-off frequency, f_1. In order to eliminate this noise, even when using an imperfect filter, we should employ a non-ideal ADM decoder.

Fig. 3. PSD of n(t) and $n_0(t)$



NON-IDEAL ADM DECODER

To construct a non-ideal ADM decoder we replace the ideal accumulator, represented by Eq.(3), by a leaky accumulator described by

$$\hat{x}(k) = \beta\ \hat{x}(k-1) + s_x(k), \tag{28}$$

where β, the leak factor, is less than unity. We should note that the realization of any practical ADM system will always use a leaky accumulator. Since the response of the non-ideal ADM decoder to a step input is an exponentially decaying function, the D.C. portion of E(k) will take the form of $n_0(t)$, shown in Fig. 2. Observing $n_0(t)$ we see that the use of a non-ideal ADM decoder guarantees that the D.C. noise offsets are non-cumulative.

In calculating the PSD of $n_0(t)$, we realize that $n_0(t)$ can be considered as the output of a leaky integrator whose input is n'(t). Using the transfer function of a leaky integrator and Eq.(26), the PSD of $n_0(t)$ is seen to be

$$G_{n_o}(f) = [pf_s/100(\alpha^2+4\pi^2 f^2)]\ \overline{I^2}, \tag{29}$$

where α is the leak factor of the leaky integrator. The shape of $G_{n_o}(f)$ is shown in Fig. 3.

Studying the effect of the non-ideal ADM decoder in the frequency domain, we observe that the area under $G_{n_o}(f)$ around the zero frequency is now finite. Thus, even if the band pass filter is not perfect, the amount of noise from the low frequency stop band at the filter output will be negligible. Furthermore, as indicated by the shaded area in Fig. 3, the in-band D.C. noise power is also decreased with a non-ideal ADM decoder. In the next section we will mathematically demonstrate the beneficial effect of the leaky accumulator on the D.C. noise levels.

A.C. PORTION OF THE NOISE WAVEFORM

The second term in Eq.(20) represents the A.C. portion of E(k) for the case of multiple bit errors. The fixed step size of the linear decoder indicated by this term, is a multiple of S. The linear decoder output for any value of a_n is a scaled version of its output when $a_n=4$. A similar situation holds for the corresponding frequency spectra.

We have experimentally verified that when an ADM encoded signal is decoded with a fixed step size or linear DM (LDM), the resulting filtered estimate will have the same shape as the original encoded signal. In addition, the frequency spectrum of the LDM decoded signal is a scaled version of the spectrum of the ADM decoded signal.

Since $\hat{x}_e(k)$ is the sum of $\hat{x}(k)$ and E(k), its spectrum is a scaled version of the $\hat{x}(k)$ spectrum. Thus, even with the presence of random channel errors, the ADM decoded signal maintaines its shape in both time and frequency domains and therefore its inteligibility. The only effect is a change in amplitude.

The above statements are valid after we have eliminated the D.C. offsets by using the appropriate band pass filter and the leaky accumulator. The presence of the leaky accumulator slightly changes the form of Eq.(20). Employing Eq.(28), the expression for the error signal becomes

$$E(k) = \beta^{k-j_n-2}E(j_n+2) + a_n S \sum_{m=j_n+3}^{k} \beta^{k-m} e_x(m-1),\quad j_n+3 \le k \le j_{n+1}. \tag{30}$$

The first term of Eq.(30) indicates the beneficial effect of the leaky accumulator on the D.C. portion of E(k). The second term indicates that the linear DM decoder, which generates the A.C. portion of E(k), is now a non-ideal and is described by

$$\hat{x}(k) = \beta\hat{x}(k-1) + S'e_x(k-1), \tag{31}$$

where $S'=a_nS$. The difference in decoding an ADM bit stream with an ideal or a non-ideal LDM decoder is negligible, as long as the original encoded signal has zero D.C. level.

THE RANGE OF a_n

For statistically independent random channel errors, the coefficient

a_n, from Eqs.(24) and (25), can be expressed as

$$a_n = \sum_{i=0}^{n} x_i , \tag{32}$$

where the random variable x_0 is zero and the random variables x_i, $i=1,2,\dots,n$, take on values 0, +4 and -4 with equal probability. Based on the Central Limit theorem it can be proven [5] that for large n the probability density function of a_n is given by

$$f_{a_n}(a_n) \cong (\sqrt{3}/2\sqrt{n\pi})\, e^{-3a_n^2/64n} \sum_k \delta(a_n-4k), \quad k=..,-1,0,1,.. \quad . \tag{33}$$

Table 1 shows the values of n for a number of average channel error rates and signal durations. It is assumed that f_s=32 KHz. For these values of n, using Eq.(33), we calculate a_n(max) when the area under $f_{a_n}(a_n)$, evaluated from $-a_n$(max) to $+a_n$(max), is equal to 0.9 and 0.95. Table 1 shows the corresponding values of a_n(max).

Table 1. Values of n and a_n (max)

Channel Error Rate	Signal Duration	a_n(max)		n
		Confidence 90%	95%	
10^{-5}	5 min	52	64	100
	10 min	76	92	200
10^{-4}	5 min	176	212	1000
	10 min	276	400	2000

We wish to keep a_n(max) as small as possible. Growth of a_n(max) in the positive direction effects the system's performance due to the limits of the internal arithmetic. Growth in the negative direction attenuates the decoded signal and decreases the decoder step size. If this step size becomes too small, the general results obtained previously will not be valid since they were based on Eq.(5) which requires that $s_x(k-1)\neq 0$, with or without errors.

SIMULATION RESULTS

We have simulated all pertinent ADM systems, with and without channel errors, on a digital computer. In this computer simulation the final output filter was realized as a band pass, finite impulse response, digital filter whose upper and lower cut-off frequencies were 3.1 KHz and 400 Hz respectively. The filter frequency response, $H_F(f)$, corresponding to 100 points of its symmetric impulse response, is shown in Fig. 4. We also have used S=0.04 V and $f_s=1/T_s$=32 KHz.

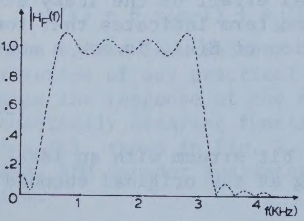


Fig. 4. Final Output Filter Transfer Function

Our first test signal was a single tone sinusoid of constant amplitude and frequency set at 800 Hz. In Fig. 5 we show one period of the filter output after we have introduced an error resulting in an a_n=4, with and without a leaky accumulator (β=0.98 and β=1). We see that when β=1 the output contains a D.C. component. All ensuing results use a leaky accumulator (β=0.98). In Fig.6 we present one period of the filtered output for a_n=0, -4, +4 and

+20. We observe that the corresponding amplitudes are such that they directly relate to the appropriate a_n values.

Our second test signal was a triangular periodic waveform. In Fig. 7 we show one period of the filtered output for a_n=0, +8 and -4. In Fig. 8 we present the amplitude spectra for these triangular waveforms and the ratio of each spectrum to the error free spectrum. The average value of each ratio waveform is

$$\left|\frac{\overline{\hat{X}_e(f)}}{\hat{X}(f)}\right|_{a_n=-4} = \left|\overline{\frac{\hat{X}(f)-N(f)}{\hat{X}(f)}}\right|_{a_n=-4}$$

$$= 1 - \left|\overline{\frac{N(f)}{\hat{X}(f)}}\right|_{a_n=-4} = 0.75 \qquad (34)$$

and

$$\left|\frac{\overline{\hat{X}_e(f)}}{\hat{X}(f)}\right|_{a_n=8} = \left|\overline{\frac{\hat{X}(f)-N(f)}{\hat{X}(f)}}\right|_{a_n=8}$$

$$= 1 + \left|\overline{\frac{N(f)}{\hat{X}(f)}}\right|_{a_n=8} = 1.46. \qquad (35)$$

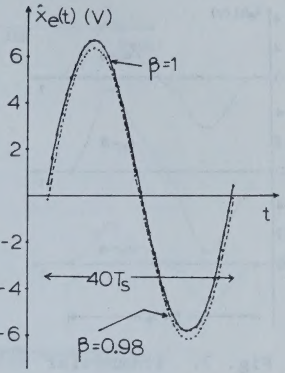


Fig. 5. Filtered output for a_n=4 with a single tone input

Here N(f) represents the frequency spectrum of the A.C. portion of E(k). Equations (34) and (35) yield

$$\left|\overline{N(f)}\right|_{a_n=-4} = 0.25\left|\overline{\hat{X}(f)}\right| \qquad (36)$$

and

$$\left|\overline{N(f)}\right|_{a_n=8} = 0.46\left|\overline{\hat{X}(f)}\right|. \qquad (37)$$

The ratio of the two averages given by Eqs.(36) and (37) is equal to 1.84 which is fairly close to the expected value (8/4=2). In the above calculation, we have tacitly assumed that the phase spectra will cancel. This manipulation is supported by our simulation results in the time domain.

Our last test signal was a pseudorandom periodic sequence, obtained by passing the output of a random noise generator through a low pass filter with cut-off frequency of 5 KHz. In Fig. 9 we show one period of the filtered output with no errors (a_n=0).

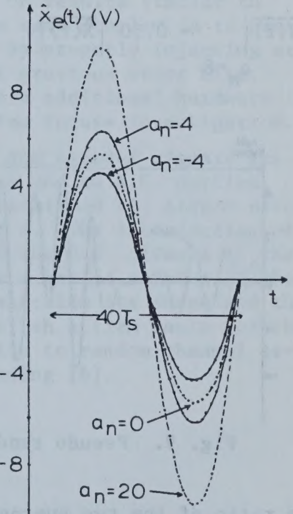


Fig. 6. Sinewave output with errors

In the same figure we present portions of the period for $a_n=16$ and 24. Figure 10 shows the frequency spectra and ratios for this test signal with and without errors. The average values of the spectra ratios are now 1.25 for $a_n=+4$ and 1.5 for $a_n=+8$.

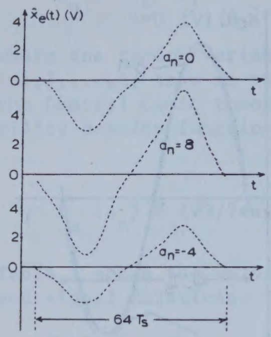


Fig. 7. Triangular output with errors

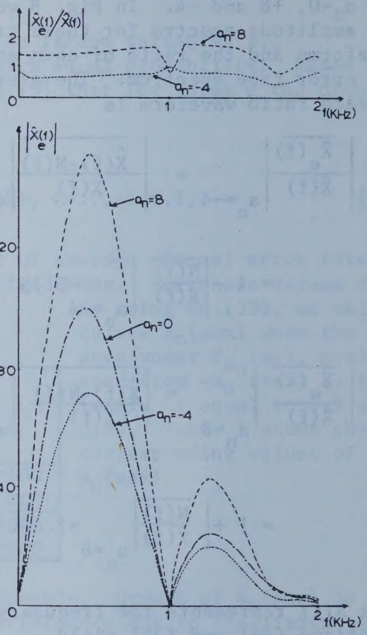


Fig. 8. Amplitude spectra of the triangular outputs with errors

For this example, we have

$$\left|\overline{N(f)}\right|_{a_n=4} = 0.25\left|\overline{\hat{X}(f)}\right| \qquad (38)$$

and

$$\left|\overline{N(f)}\right|_{a_n=8} = 0.50\left|\overline{\hat{X}(f)}\right|. \qquad (39)$$

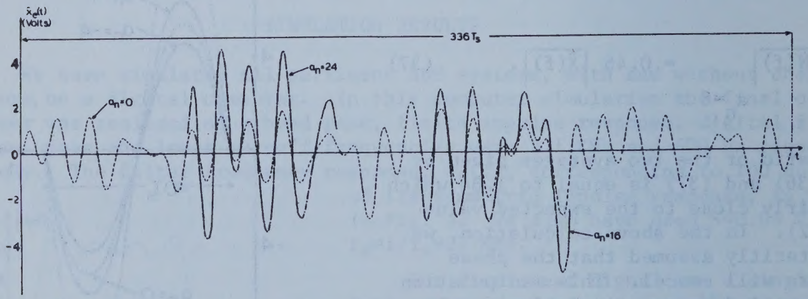


Fig. 9. Pseudo random sequence with and without errors

The ratio of the two averages given in Eqs. (38) and (39) is equal to 2.0 which is exactly the expected value (8/4=2).

All results given above verify our theoretical developments for the behavior of an ADM encoded signal in the presence of random channel errors.

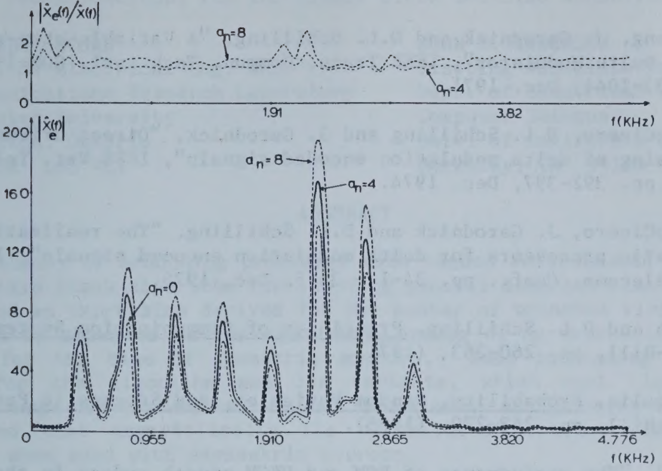


Fig. 10. Amplitude spectra of the pseudo-random sequence output with and without errors

IMPROVEMENT TECHNIQUES/CONCLUSIONS

We already have stated the effects of large excursions of a_n in the positive or negative direction. To protect the system we can simply increase the size of the internal arithmetic based on results similar to those shown in Table 1. Another way to fight the same problem is to prevent the large negative or positive values of a_n by properly injecting artificial error bits so as to cancel the effect of previous error bits. This will result in an extra time delay and require additional hardware to realize a decision making system. This is left for future investigation.

We have shown that random channel errors in ADM encoded signals generate a D.C. offset which is cummulative in nature and an A.C. portion which is obtained by linearly DM decoding the transmitted bit stream using a fixed step size which is an integer multiple of S. The introduction of a leaky accumulator in the ADM decoder eliminates the D.C. offsets of the noise signal. The remaining A.C. noise signal is a scaled version of the input signal. As a result, the decoded signal maintains its shape and thus its inteligibility. Our simulation results establish a firm basis for the claim that ADM encoded signals have a good immunity to random channel errors when compared to other forms of digital encoding [6].

REFERENCES

[1] C.L. Song, J. Garodnick and D.L. Schilling, "A Variable step-size Robust Delta Modulator", IEEE Trans. Commun. Tech. vol. COM-19, pp. 1033-1044, Dec. 1971.

[2] J.L. LoCicero, D.L. Schilling and J. Garodnick, "Direct arithmetic processing of delta modulation encoded signals", IEEE Nat. Telecomm. Conf., pp. 392-397, Dec. 1974.

[3] J.L. LoCicero, J. Garodnick and D.L. Schilling, "The realization of arithmetic processors for delta modulation encoded signals", IEEE Nat. Telecomm. Conf., pp. 34-1 - 34-5, Dec. 1975.

[4] H. Taub and D.L. Schilling, Principles of Communication Systems, McGraw-Hill, pp. 260-263, (1971).

[5] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, pp. 268-270, (1965).

[6] P. Noll, "The performance of PCM and DPCM speech coders in the presence of independent and correlated errors", IEEE Int. Conf. on Commun., vol. II, pp. 30-1 - 30-5, June 1975.

BRANCHING PROCESS METHODS FOR THE SINGLE STACK ENCODING ALGORITHM

SESHADRI MOHAN
Dept. of Electrical Eng. and
Communications Research Laboratory
McMaster University
Hamilton, Ontario
Canada L8S 4L7

JOHN B. ANDERSON
Visiting Associate Professor
Dept. of Electrical Eng. and
Computer Science
Univ. of California at Berkeley
Berkeley, CA 94720 U.S.A.

ABSTRACT

The theory of branching processes with random environments is applied to the single stack algorithm for encoding general discrete-time memoryless sources and an expression derived for the number of branches visited. This expression is shown to be the stochastic analog of an expression given by Gallager for the case of symmetric sources. Some intriguing simulation results for the algorithm and its variants, which cast light on the possible range of solutions to our equations, are presented. It is conjectured that asymptotically the algorithm's behaviour is not much different when used with asymmetric sources.

1. INTRODUCTION

Initially, interest in rate-distortion theory centered around developing rate-distortion functions for different sources. Only ten years after the birth of the theory was attention given to one of the two facets of practical source coding, the design of codes for sources, by Jelinek [1], who proved the existence of tree codes that achieve the rate-distortion bound. Jelinek's proof was valid only for symmetric sources, and it remained for Davis and Hellman [2], making use of the theory of branching processes in random environments (BPRE) [3], to prove that for any i.i.d. source, tree codes exist whose performance is as close to the R(D) curve as desired. Tan [4], again using the BPRE theory [5], showed that such tree codes exist for stationary block-ergodic sources. Since the paper by Jelinek [1] appeared in the literature, attention has also centered around the second facet of source coding, the design and analysis of source coding algorithms (see Anderson [6], Anderson and Jelinek [7], and Gallager [8]). While these papers consider only symmetric sources, our interest in this paper is in applying BPRE theory to the single stack encoding algorithhm applied to asymmetric sources, with the aim of deriving an expression for the number of tree branches visited.

2. PRELIMINARIES

2.1 Rate-Distortion Theory

Let a discrete memoryless source (i.i.d. source) have a probability distribution $P(x)$ defined on source alphabets $x \in X$. Define an additive single letter distortion measure $d(x,y)$ between source alphabets $x \in X$ and reproduction alphabets $y \in Y$. Let $Q(y|x)$ be a conditional probability assignment and $I(x;y)$ the mutual information between x and y.

Definition: The rate-distortion function R(D) is the minimum rate necessary to encode the information source with additive single letter distortion measure so that the average distortion does not exceed some distortion D; it is given by

$$R(D) \overset{\Delta}{=} \inf_{Q \in Q_D} I(x;y) \tag{1}$$

where

$$Q_D = \{Q(y|x): \sum_{x,y} P(x)Q(y|x)d(x,y) \leq D\} \quad . \tag{2}$$

Haskell [9] has shown that

$$R(D) = \max_{\rho \geq 0} \min_{Q(y)} - \sum_{x} P(x) \log \{\sum_{y} Q(y) \exp [-\rho(d(x,y)-D)]\} \tag{3}$$

where $Q(y)$ is a probability density defined on the reproduction alphabet given by

$$Q(y) = \sum_{x} P(x) Q(y|x) \quad . \tag{4}$$

Let ρ_o and $Q_o(y)$ optimize the expression in (3). Then we have the following equations [10, pp. 34-37].

$$Q_o(y|x) = Q_o(y) \exp [-\rho_o d(x,y)] \lambda(x) \tag{5}$$

$$\lambda(x) = \{\sum_{y} Q_o(y) \exp [-\rho_o d(x,y)]\}^{-1} \tag{6}$$

$$Q_o(y) = \sum_{x} P(x) Q_o(y|x) \tag{7}$$

$$R(D) = -\rho_o D + \sum_{x} P(x) \log \lambda(x) \quad . \tag{8}$$

2.2 Branching Processes with Random Environments

We will find in 3 that branching processes with random environments (BPRE), formulated by Smith and Wilkinson [3] and extended by Athreya and Karlin [5], are an appropriate tool for analyzing the single stack encoding algorithm. So we briefly describe the working of a BPRE. Harris [11] and Athreya and Ney [12] are excellent sources on the theory of branching processes.

Suppose we have Z_o particles at time $n = 0$ (or generation 0). Each of these Z_o particles creates further particles so that the population size at the first generation is

$$Z_1 = \sum_{i=1}^{Z_o} X_{1i} \tag{9}$$

where X_{1i}, $i = 1, 2, \ldots, Z_o$ are independent and identically distributed random variables with probability generating function (p.g.f.) $\phi_{\zeta o}(s)$. Here

$$\phi_{\zeta o}(s) = \sum_{j=0}^{\infty} p_{\zeta o}(j) \, s^j \tag{10}$$

where $p_{\zeta o}(j)$ is the probility that a zeroth generation particle gives rise to j first generation particles, given environment ζ_o. The Z_1 first generation particles then give rise to second generation particles according to p.g.f. $\phi_{\zeta 1}(s)$. Continuing in this way, the (n+1)th generation population is the cumulative sons or progenies of the Z_n nth generation particles, each reproducing according to p.g.f. $\phi_{\zeta n}(s)$. $\{\zeta_n, n = 0, 1, 2, \ldots\}$ is called the *environmental process*. In the case of an i.i.d. environmental process, $\{\phi_{\zeta n}(s)\}$ are all identical, $\{\zeta_n\}$ are i.i.d., and Z_n, $n = 0, 1, 2, \ldots$ is a branching process developing in an i.i.d. random environment. We can visualize the Z_n process as one developing in a stochastically changing environment that affects the reproductive behaviour of the process.

Allowing for generalization, we now stipulate that a particle may be any one of a number of *types*, say m types. Thus, starting from a zeroth generation particle of type i, we have at the first generation γ_1 particles of type 1, γ_2 of type 2, ..., γ_m of type m produced according to p.g.f.

$$\phi^{(i)}_{\zeta o}(\underset{\sim}{s}) = \sum_{\gamma_1,\gamma_2,\ldots,\gamma_m} p^{(i)}_{\zeta o}(\gamma_1,\gamma_2,\ldots,\gamma_m)\, s_1^{\gamma_1} s_2^{\gamma_2} \ldots s_m^{\gamma_m} \qquad (11)$$

where $\underset{\sim}{s} = (s_1, s_2, \ldots, s_m)$, $0 \leq s_i \leq 1$, $i = 1, \ldots, m$. Again a type j particle at the first generation reproduces according to p.g.f. $\phi^{(j)}_{\zeta 1}(\underset{\sim}{s})$, and so on. Thus we have a vector process defined by $\underset{\sim}{Z}_o = \underset{\sim}{e}_i = (0, 0, \ldots, 1, \ldots, 0)$, a vector with 1 in the ith place and 0's elsewhere, and $\underset{\sim}{Z}_1 = (Z_1(i,1), Z_1(i,2), \ldots, Z_1(i,m))$ etc, where $Z_1(i,\gamma)$ is the number of first generation particles of type γ from a zeroth generation particle of type i. In general, if $\underset{\sim}{Z}_n = (Z_n(i,1), Z_n(i,2), \ldots, Z_n(i,m))$, then the (n+1)th generation population vector $\underset{\sim}{Z}_{n+1}$ is the sum of $Z_n(i,1) + Z_n(i,2) + \ldots + Z_n(i,m)$ independent random vectors, where each of the $Z_n(i,k)$, $k = 1, 2, \ldots, m$, independent random vectors is produced according to the probability assignment $p^{(k)}_{\zeta n}(\underset{\sim}{\gamma})$, where $\underset{\sim}{\gamma}$ is an m vector. Again, when $\{\zeta_n\}$ is i.i.d., $\underset{\sim}{Z}_n$, $n = 0, 1, \ldots$, is a multitype (m-type) branching process in random environment.

The above process is said to be extinguished if $\underset{\sim}{Z}_n = \underset{\sim}{0}$ for some n, where $\underset{\sim}{0} = (0, 0, \ldots, 0)$ is an m vector. If the extinction probability $q^{(i)}$ of the process, starting from an initial particle of type i, is defined as

$$q^{(i)} \overset{\Delta}{=} P[\underset{\sim}{Z}_n = \underset{\sim}{0} \text{ for some } n \mid \underset{\sim}{Z}_o = \underset{\sim}{e}_i], \qquad (12)$$

then we can associate a probability vector q with the process, given by $q = (q^{(1)}, q^{(2)}, \ldots, q^{(m)})$. If $\underset{\sim}{\zeta} = (\zeta 0, \zeta 1, \ldots)$, then from [5] $\underset{\sim}{q}(\underset{\sim}{\zeta})$, the extinction probability conditioned on the environment $\underset{\sim}{\zeta}$, satisfies the functional equation

$$\underset{\sim}{q}(\underset{\sim}{\zeta}) = \lim_{n\to\infty} \underset{\sim}{\phi}_{\zeta 0}(\underset{\sim}{\phi}_{\zeta 1}(\ldots \underset{\sim}{\phi}_{\zeta n}(\underset{\sim}{s}))\ldots)) \qquad (13)$$

$$= \underset{\sim}{\phi}_{\zeta o}(\underset{\sim}{q}(T\underset{\sim}{\zeta})) \qquad (14)$$

where T is the shift operator and $T\underset{\sim}{\zeta} = (\zeta 1, \zeta 2, \ldots)$ and the vector $\underset{\sim}{\phi}$ is given by $\underset{\sim}{\phi}_{\zeta\gamma}(\underset{\sim}{s}) = (\phi^{(1)}_{\zeta\gamma}(\underset{\sim}{s}), \ldots, \phi^{(m)}_{\zeta\gamma}(\underset{\sim}{s}))$. $\underset{\sim}{q}$, the unconditional probability of extinction, is then $\underset{\sim}{q} = E[\underset{\sim}{q}(\underset{\sim}{\zeta})]$.

While the papers [3] and [5] deal with conditions for certain and noncertain extinction and with proving certain limit theorems, our interest here is to model the single stack encoding algorithm as a multi-type BPRE and to derive an expression for the number of branches visited.

2.3 Tree Codes

Consider random tree codes with b branches out of each node and β symbols on each branch. The symbols are chosen from reproduction alphabet set Y according to the distribution $Q_o(y)$ given in (7). The rate R of such a tree code is given by

$$R \overset{\Delta}{=} \frac{\log b}{\beta} \quad . \tag{15}$$

If D_o is the desired average distortion at the end of encoding then R is chosen to satisfy $R \geq R(D_o)$, where R() is defined in (8). Given any $\varepsilon > 0$, the object of encoding a given source is to find a path through the code tree that has an average distortion $D \leq D_o + \varepsilon$. Associated with a path $\underset{\sim}{y}^{\ell}$ of ℓ/β branches is a quantity called the metric of the path defined by

$$\mu(\underset{\sim}{y}^{\ell}) \overset{\Delta}{=} D(\underset{\sim}{x}^{\ell}) - d(\underset{\sim}{x}^{\ell}, \underset{\sim}{y}^{\ell}) \tag{16}$$

where $D(\underset{\sim}{x}^{\ell})$ is defined as

$$D(\underset{\sim}{x}^{\ell}) \overset{\Delta}{=} \sum_{\underset{\sim}{y}^{\ell}} d(\underset{\sim}{x}^{\ell}, \underset{\sim}{y}^{\ell})\, Q_o(\underset{\sim}{y}^{\ell} | \underset{\sim}{x}^{\ell}). \tag{17}$$

Here $\underset{\sim}{x}^{\ell}$ and $\underset{\sim}{y}^{\ell}$ refer to length ℓ source and reproduction symbols respectively. Also, if

$$\underset{\sim}{x}^{\ell} = (x_{11}, x_{12}, \ldots, x_{1\beta}, x_{21}, \ldots, x_{2\beta}, \ldots, x_{\ell/\beta,1}, \ldots, x_{\ell/\beta,\beta})$$

$$= (\underset{\sim}{x}^{\beta}_{1}, \underset{\sim}{x}^{\beta}_{2}, \ldots, \underset{\sim}{x}^{\beta}_{\ell/\beta})$$

and $\underset{\sim}{y}^{\ell}$ is similarly defined, then

$$D(\underset{\sim}{x}^{\ell}) = \sum_{i=1}^{\ell/\beta} D(\underset{\sim}{x}^{\beta}_{i}) \tag{18}$$

and

$$d(\underset{\sim}{x}^{\ell}, \underset{\sim}{y}^{\ell}) = \sum_{i=1}^{\ell/\beta} d(\underset{\sim}{x}^{\beta}_{i}, \underset{\sim}{y}^{\beta}_{i}) . \tag{19}$$

Thus

$$\mu(\underset{\sim}{y}^{\ell}) = \sum_{i=1}^{\ell/\beta} \mu(\underset{\sim}{y}^{\beta}_{i}) \quad . \tag{20}$$

Here the $d(\underset{\sim}{x}^{\beta}_{i}, \underset{\sim}{y}^{\beta}_{i})$'s and $\mu(\underset{\sim}{y}^{\beta}_{i})$'s are branch distortion values and branch metric increments, respectively.

In defining the metric of a code tree path, following Davis and Hellman [2], we have used the $D(\underset{\sim}{x}^{\ell})$ criterion instead of the traditional one using D_o. Next, in order to model the algorithm as a multitype BPRE, we quantize the metric values of paths. Where no confusion will arise, we will still call the quantized metric as metric only and denote it by the same symbol μ. Thus, redefining the metric, we have

$$\mu(\underset{\sim}{y}^{\ell}) = \frac{1}{\xi}\left\lceil D(\underset{\sim}{x}^{\ell}) - d(\underset{\sim}{x}^{\ell}, \underset{\sim}{y}^{\ell}) \right\rceil \tag{21}$$

where $\lceil a \rceil$ is the smallest integer N such that $a \leq N$ and ξ is an arbitrary small positive number. Still,

$$\mu(\underset{\sim}{y}^{\ell}) = \sum_{i=1}^{\ell/\beta} \mu(\underset{\sim}{y}_i^{\beta}).$$

Next we describe the encoding algorithm followed by an application of BPRE methods to it.

3. ANALYSIS OF THE SINGLE STACK ENCODING ALGORITHM

The single stack algorithm is described in detail in [13] and [14]. Gallager [8] proposes a similar algorithm. However, we give below a brief description. We assume that the algorithm operates on the tree codes described in 2.3. In the following, we mean by a "completely searched node" that all branches out of that node have been explored.

0) Initially, the path map corresponding to the root node (with metric 0) resides in the stack. Also, a save area initially contains the root node.

1) From the topmost node in the stack, extend a branch not yet visited. Push down the new node onto the stack and compute its metric.

2) If the metric falls below a lower barrier B, go to 3). Otherwise go to 4).

3) If the path in the stack is longer than that in the save area, store it in the save area. Pop nodes (i.e. backtrack) from the stack until a node that is not yet completely searched is found. If the stack becomes empty in the process, release the path in the save area, and reinitialize it and the stack.

4) If a stopping rule is satisfied, release either the path in the stack or that in the save area, whichever is longer, and stop. Otherwise go to 1).

The above algorithm is essentially a depth-first search procedure, trying to explore along the depth of the code tree a path that satisfies a discard criterion. When the path falls below barrier B, the algorithm backtracks along the path to another node from where the search again proceeds in a depth-first fashion. For the purposes of analysis we introduce an upper barrier A and consider only paths that lie within the barriers B and A. Because of the definition in (21), μ is now allowed to take only integer values lying between B and A. There are (A-B+1) such metric values. We denote the set of allowed metric values as N(B,A). Introduction of upper barrier A will not appreciably alter the behaviour of the above algorithm if A is considered to be very large. We assume further that the algorithm works successively on source blocks of ℓ symbols.

Now we turn to a BPRE model of the above algorithm. At any stage during encoding we can consider code tree paths as particles and their metrics as the particle types. Thus we have an $m = (A-B+1)$ type process. For the symmetric source case, all source blocks of length ℓ correspond to the same environment. In the case of asymmetric sources this is no longer true. Thus if a block of length ℓ source sequence is an atypical one (i.e. a hostile environment) it may be particularly difficult to find a code tree path of length ℓ/β branches that lies within the barriers. In such a case the algorithm may explore a large number of code tree branches. If one were to consider paths of length ℓ/β branches whose metrics belong to the set $N(B,A)$ as offspring of the root node, then it is clear that the distribution of offspring varies from generation to generation (i.e. from one ℓ-block to another ℓ-block). Hence the behaviour of the algorithm within a block is dependent on the source sequence $\underset{\sim}{x}_{\ell}$ corresponding to that block. The "environment" within a block can be identified with $\underset{\sim}{x}_{\ell}$, and successive environments vary in an i.i.d. manner, since $\underset{\sim}{x}_{\ell}$'s are i.i.d. Thus we have an $m = (A-B+1)$ type BPRE.

The single-type BPRE formulation of Jelink's tree coding process was first given by Davis and Hellman [2], and we have analysed a new coding process, the single stack algorithm, in a similar manner, but as a multi-type BPRE. Our asymmetric case analysis is similar in form to the one employed by Gallager [8] for the symmetric case, and we have used several of his results.

A node is said to be visited when it is pushed down onto the stack (step 1 of the algorithm). Define the node computation as the toal number of nodes visited. A branch may be traversed either in the forward direction when a node is pushed onto the stack (step 1), or in the reverse direction when a node is popped up from the stack (step 3). Any branch may at most be traversed twice. Thus the total number of branches visited is upperbounded by twice the node computation. Define now the following quantities.

$\alpha^{(j-i)}(\underset{\sim}{x}^{\ell}) \triangleq P$ {an immediate descendent of a node with metric i has metric j, given that the environment is $\underset{\sim}{x}^{\ell}$}.

$q_{\gamma}^{(i)}(\underset{\sim}{x}^{\ell}) \triangleq P$ {no γ-th level descendent of a node with metric i has metric $j \in N(B,A)$, given environment $\underset{\sim}{x}^{\ell}$}.

This is the same as the probability that the process forward of a node with metric i is extinguished at or before γth level, given environment $\underset{\sim}{x}^{\ell}$.

$\Lambda \triangleq \ell/\beta$.

$C_{\Lambda}^{(i)}(\underset{\sim}{x}^{\ell}) \triangleq E$ {node computation forward of a node with metric i, in order to encode a length ℓ source seuqence or get extinguished, given environment $\underset{\sim}{x}^{\ell}$}.

Theorem 1

Let the barriers be at B and A. Let the algorithm start with a root node with metric i and encode a source sequence $\underset{\sim}{x}^{\ell}$. Then for $\Lambda \geq 2$,

$$C_\Lambda^{(i)}(\underset{\sim}{x}^\ell) = 1 + U_\Lambda^{(i)}(\underset{\sim}{x}^\ell) + U_\Lambda^{(i)}(\underset{\sim}{x}^\ell)\,\rho_\Lambda^{(i)}(\underset{\sim}{x}^\ell) +$$

$$U_\Lambda^{(i)}(\underset{\sim}{x}^\ell)\,\{\rho_\Lambda^{(i)}(\underset{\sim}{x}^\ell)\}^2 + \ldots + U_\Lambda^{(i)}(\underset{\sim}{x}^\ell)\,\{\rho_\Lambda^{(i)}(\underset{\sim}{x}^\ell)\}^{b-1} \qquad (22)$$

where

$$U_\Lambda^{(i)}(\underset{\sim}{x}^\ell) = \sum_{j\in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^\ell)\; C_{\Lambda-1}^{(j)}(\underset{\sim}{x}^\ell) \qquad (23)$$

and

$$\rho_\Lambda^{(i)}(\underset{\sim}{x}^\ell) = 1 - \sum_{j\in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^\ell)\;\{1 - q_{\Lambda-1}^{(j)}(\underset{\sim}{x}^\ell)\}\ . \qquad (24)$$

Proof: For the symmetric case the above results have been proved by Gallager [8]. The same method applies to asymmetric sources when conditioned on $\underset{\sim}{x}^\ell$.

Note that the result for asymmetric sources is the stochastic analog of the result for symmetric sources.

Lemma 1

$$q_\gamma^{(j)}(\underset{\sim}{x}^\ell) \leq q_\eta^{(j)}(\underset{\sim}{x}^\ell) \quad \text{for} \quad 0 \leq \gamma \leq \eta \leq \Lambda\ . \qquad (25)$$

Proof: See Harris [11, p. 7].

Lemma 2

Define

$$S(\delta,\ \underset{\sim}{x}^\ell) \overset{\Delta}{=} \{\underset{\sim}{y}^\ell : \mu(\underset{\sim}{y}^\ell) \geq -\frac{\ell\delta}{\xi}\} \qquad (26)$$

and

$$P^{(j)}(\underset{\sim}{x}^\ell) \overset{\Delta}{=} \sum_{S(\delta,\underset{\sim}{x}^\ell)} Q_o(\underset{\sim}{y}^\ell) \qquad (27)$$

where j is the metric of the root node. Assume that D_m is the maximum distortion such that $d(x,y) \leq D_m$, where $x \in X$ and $y \in Y$. Then

$$P^{(j)}(\underset{\sim}{x}^\ell) \geq \exp(-\rho_o j)\;[\lambda(\underset{\sim}{x}^\ell)]^{-1}\;(1 - D_m^2/\delta^2\ell)$$

$$.\ \exp\,[\rho_o\;(D(\underset{\sim}{x}^\ell) - \ell\delta)]\ . \qquad (28)$$

Proof: See Davis and Hellman [2]. Their results are only slightly modified here.

Lemma 3

If $B = -\lfloor \ell\delta/\xi \rfloor$ where $\lfloor a \rfloor$ is the largest integer N such that $a \geq N$, then

$$q_{\Lambda-1}^{(j)}(\underset{\sim}{x}^\ell) \leq 1 - P^{(j)}(\underset{\sim}{x}^\ell)\ . \qquad (29)$$

Proof: From lemma 1, $q^{(j)}_{\Lambda-1}(\underset{\sim}{x}^{\ell}) \leq q^{(j)}_{\Lambda}(\underset{\sim}{x}^{\ell})$. If the process fails to find a node at level Λ with metric $k \in N(B,A)$ then the first descendent $\underset{\sim}{y}^{\ell}_{1}$ at level Λ from the root node with metric j is also not in N(B,A). Therefore

$$q^{(j)}_{\Lambda}(\underset{\sim}{x}^{\ell}) \leq P\{\mu(\underset{\sim}{y}^{\ell}_{1}) \notin N(B,A) \mid \text{root node metric} = j\}$$

$$= P\{\underset{\sim}{y}^{\ell}_{1} \notin S(\delta,\underset{\sim}{x}^{\ell}) \mid \text{root node metric} = j\} \tag{30}$$

$$= 1 - P^{(j)}(\underset{\sim}{x}^{\ell}) \ . \tag{31}$$

Theorem 3

$$C^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell}) \leq 1 + U^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell}) \frac{[1-\{\Gamma^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell})\}^{b}]}{[1-\Gamma^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell})]} \tag{32}$$

where

$$\Gamma^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell}) \triangleq 1 - [\lambda(\underset{\sim}{x}^{\ell})]^{-1} (1 - D^{2}_{m}/\delta^{2}\ell) \exp[\rho_{o}(D(\underset{\sim}{x}^{\ell})-\ell\delta)] G^{(i)}(\underset{\sim}{x}^{\ell}) \tag{33}$$

and

$$G^{(i)}(\underset{\sim}{x}^{\ell}) \triangleq \sum_{j \in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^{\ell}) \exp(-\rho_{o}j) \ . \tag{34}$$

Proof: From theorem 1

$$\rho^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell}) = 1 - \sum_{j \in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^{\ell}) \{1 - q^{(j)}_{\Lambda-1}(\underset{\sim}{x}^{\ell})\} \tag{35}$$

$$\leq 1 - \sum_{j \in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^{\ell}) \{1 - q^{(j)}_{\Lambda}(\underset{\sim}{x}^{\ell})\} \tag{36}$$

$$\leq 1 - \sum_{j \in N(B,A)} \alpha^{(j-i)}(\underset{\sim}{x}^{\ell}) P^{(j)}(\underset{\sim}{x}^{\ell}) \tag{37}$$

$$\leq 1 - [\lambda(\underset{\sim}{x}^{\ell})]^{-1} (1 - D^{2}_{m}/\delta^{2}\ell) \exp[\rho_{o}(D(\underset{\sim}{x}^{\ell}) - \ell\delta)] G^{(i)}(\underset{\sim}{x}^{\ell}) \tag{38}$$

$$= \Gamma^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell}) \ . \tag{39}$$

(36) follows form lemma 1. The next step follows from lemma 3. Applying the inequality in lemma 2 and the definition of $G^{(i)}(\underset{\sim}{x}^{\ell})$ in (33) to (37) results in the next step. Bounding $\rho^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell})$ in the expression for $C^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell})$ in (22) by $\Gamma^{(i)}_{\Lambda}(\underset{\sim}{x}^{\ell})$ and doing a simple manipulation on the resulting expression yields the theorem.

4. CONCLUSIONS AND SIMULATIONS

For the purpose of simulation, we have used the binary i.i.d. source and Hamming distortion measure. The algorithm worked on the ensemble of rate-1/2 tree codes ($b = \beta = 2$) with code words chosen i.i.d. according to the output probability distribution [10, p. 37]

$$Q_o(0) = (P(0) - D_o)/(1 - 2D_o)$$

$$Q_o(1) = (P(1) - D_o)/(1 - 2D_o)$$

where D_o is given by the distortion-rate function $\Delta(R)$. The algorithm was parameterized by the lower barrier B and the depth limit or block length Λ. In all our simulation examples, the algorithm encoded a few thousand source samples and used a depth limit of 200 to 1000 tree branches. Table 1 lists simulation results which cast some light on the possible range of solutions to our equations for node computation derived in 3. Results obtained using the binary symmetric source is also given for comparison. The computation in the asymmetric case is an order of magnitude smaller than that for the symmetric case. But one should refrain from making hasty conclusions as the algorithm searched a large number of branches (of the order of a few thousand) in order to encode the binary source with $P(0) = 0.2$ and $P(1) = 0.8$. The increased search may be due to a "hostile" environment encountered by the algorithm while encoding, thus justifying the BPRE model suggested. While the algorithm seems to have a complicated dependence on various parameters, as the simulation results seem to suggest, the following question arises. Asymptotically, does the algorithm behave quite differently when used with asymmetric sources? While we have not yet proved so, we conjecture that the algorithm's behaviour is not much different when used with asymmetric sources and that the node computation C, as is the case with many different algorithms used with symmetric sources [6], [7] and [8], is of the form $C = \exp[c(D-D_o)^{-\alpha}]$, where c and α are constatants, $D_o = \Delta(R)$, and D is the expected distortion per output symbol for the encoded path.

REFERENCES

[1] F. Jelinek, "Tree encoding of memoryless time-discrete sources with a fidelity criterion", IEEE Trans. Inform. Theory, vol. IT-15, pp. 584-590, Sept. 1969.

[2] C.R. Davis and M.E. Hellman, "On tree coding with a fidelity criterion", IEEE Trans. Inform. Theory, vol. IT-21, pp. 373-378, July 1973.

[3] W. Smith and W. Wilkinson, "On branching processes in random environments", Ann. Math. Statist., vol. 40, pp. 814-827, 1969.

[4] H.H. Tan, "Tree coding of discrete-time abstract alphabet stationary block-ergodic sources with a fidelity criterion", IEEE Trans. Inform. Theory, vol. IT-22, pp. 682-690.

[5] K.B. Athreya and S. Karlin, "On branching processes with random environments: I - Extinction probabilities", Ann. Math. Statist., vol. 42, pp. 1499-1520, 1971.

[6] J.B. Anderson, "A stack algorithm for source coding with a fidelity criterion", IEEE Trans. Inform. Theory, vol. IT-20, pp. 211-226, Mar. 1974.

[7] J.B. Anderson and F. Jelinek, "A 2-cycle algorithm for source coding with a fidelity criterion", IEEE Trans. Inform. Theory, vol. IT-19, pp. 77-92, Jan. 1973.

[8] R.G. Gallager, "Tree encoding for symmetric sources with a distortion measure", IEEE Trans. Inform. Theory, vol. IT-20, pp. 65-76, Jan. 1974.

[9] B. Haskell, "The computation and bounding of rate-distortion functions", IEEE Trans. Inform. Theory, vol. IT-15, pp. 525-531, Sept. 1969.

[10] T. Berger, Rate Distortion Theory: A Mathematical Basis for Data Compression, Prentice-Hall, Englewood Cliffs, N.J.: 1971.

[11] T.E. Harris, The Theory of Branching Processes, Springer, New York: 1963.

[12] K.B. Athreya and P.E. Ney, Branching Processes, Springer, New York: 1972.

[13] J.B. Anderson and S. Mohan, "A push-down stack measure of encoding algorithm complexity", Conference Record, Intern. Conf. on Communications, Toronto, Canada, June 1978.

[14] J.B. Anderson and S. Mohan, "A systemtic analysis of cost for sequential coding algorithms", in submission, IEEE Trans. Inform. Theory, July 1978.

P(0)	P(1)	B	Branches Viewed	% above $\Delta(R)$
0.5	0.5	-5.5	1500	15.0
0.3	0.7	-1.5	355	8.1
		-3.5	368	4.5
		-7.5	176	9.2
0.2	0.8	-2.5	707	39.0

TABLE 1: Simulation results for the single stack encoding algorithm. Binary i.i.d. source with Hamming distortion, $R = 1/2$, $\Lambda = 200 - 1000$.

VOICED/UNVOICED/MIXED EXCITATION CLASSIFICATION IN SPEECH ANALYSIS

LEAH J. SIEGEL
School of Electrical Engineering, Purdue University
West Lafayette, Indiana 47907

The use of pattern classification techniques to assign the excitation function for a segment of speech to one of three classes - voiced (periodic), unvoiced (noiselike), or mixed (simultaneously periodic and noiselike) - is being examined. Such techniques have been applied successfully to the voiced/unvoiced classification problem [1,2], but this two-class model does not adequately represent voiced fricative (mixed excitation) sounds.

Classification is made by dividing a multidimensional feature space into three regions, corresponding to the three classes. The classifier is a function of the set of speech segments used in training (training set) and of the features used to make the classification. Training is accomplished using a non-parametric, non-statistical technique in which a relatively small training set is used to define the voiced/unvoiced/mixed regions in the feature space. The regions are divided by hyperplanes which minimize the distance by which misclassification occurs on the training set. For computational reasons, it is desirable to use as few features as possible to make the classification, and to minimize the size of the training set used to obtain the classifier. An iterative procedure to train the classifier therefore interleaves additions to the training and feature sets. At each iteration, the training set is augmented by the speech segments near the boundaries between the regions. The feature set is augmented by the feature which yields the greatest improvement on the current training set.

A preliminary training sequence has been conducted, with a phonetically balanced set of words spoken by one speaker available for use in the training set. Features available were those which have been used in systems employing pattern classification techniques to make the voiced/unvoiced decision. These features included the energy of the signal, zero crossings, the normalized autocorrelation coefficient at delay = 1, the linear predictive coding normalized minimum error, the first linear predictor coefficient (equal to the first sample of the cepstrum), and a ratio of high frequency energy content to low frequency energy content. In addition a product of high frequency energy content and low frequency energy content was available. Execution of the training procedure failed to yield satisfactory discrimination between mixed and non-mixed excitation sounds; moreover, on a training set of 91 boundary segments, the classifier defined by using all of the available features was unable to classify mixed excitation sounds correctly. The features adequate for voice/unvoiced discrimination are therefore not sufficient for voiced/unvoiced/mixed classification. Current work involves investigating why these features are inadequate for the identification of mixed excitation, and analyzing other features for use in this task.

References

1. L. J. Siegel, "A Procedure for Using Pattern Classification Techniques to Obtain a Voiced/Unvoiced Classifier," IEEE Trans. ASSP, to be published.
2. B. S. Atal and L. R. Rabiner, "A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Applications to Speech Recognition," IEEE Trans. ASSP, Vol. 24 (June 1976), pp. 201-212.

This work was supported by NSF under Grant ENG78-05789.

AN ADAPTIVE NOISELESS VARIABLE-TO-FIXED ENCODING ALGORITHM FOR DISCRETE ERGODIC SOURCES

LEONARDUS TH. N. M. MULDER AND DAVID L. COHN
University of Notre Dame
Department of Electrical Engineering
Notre Dame, Indiana 46556

ABSTRACT

A source-coding algorithm in which the messages adapt themselves to the varying statistics of the source is described. It retains the advantages of variable-to-fixed coding and tracks changing source-statistics, thereby being a useful tool for learning unknown sources. Its performance is compared with that of two other coding techniques, viz. a fixed encoding tree algorithm and a "state-estimation" technique for a specific source.

I. INTRODUCTION

Several source-coding algorithms for memoryless sources have been developed in the literature[1,2,3,4], but their applications to time-varying sources have been mentioned only sporadically. Gallager[5] pointed out that Huffman codes can be used for sources with slowly varying statistics and showed a possible implementation of such an 'adaptive' code. However, he did not compare his technique with alternative approaches.

In this paper we will describe an adaptive variable-to-fixed algorithm, based on Tunstall's optimal variable-to-fixed algorithm for constant memoryless sources. To obtain an indication of its performance, we have compared it with two alternative variable-to-fixed algorithms in extensive simulations. The source-model used in the simulations is a variant of the Gilbert-source[8], a classical example of an ergodic non-unifilar source.

II. BACKGROUND

The purpose of noiseless source coding is to minimize the rate R, which is the average number of channel symbols per source symbol, such that the source date can be fully reconstructed. Cohn[3] introduced the class of block-to-block codes, which are characterized by an encoding-tree and a decoding-tree representing respectively the set of messages and the set of code words. In this paper we will restrict ourselves to the subclass of variable-to-fixed full-tree codes, in which the messages have various lengths and the codewords have a fixed length, while there is one unique codeword associated with every message.

Under the constraint of fixing the number of codewords n, an optimal variable-to-fixed algorithm for constant discrete memoryless sources has been described by Tunstall[1], Jelinek and Schneider[2] and Cohn[3]. The optimal set of messages is found by successively extending in the encoding-tree the node with highest probability until the number of terminal nodes equals n. For a binary memoryless source with probability p of sending a 1, in the case $p = 0.30$ and $n = 4$, this procedure is illustrated in Figure 1.

In the case that the source is time-varying optimality of above technique is no longer guaranteed, as will soon become clear. We will present an adaptive algorithm that for cases of interest outperforms a fixed-tree algorithm (based on an estimate of $\overline{p}$) as well as a state-estimation technique (see section 5).

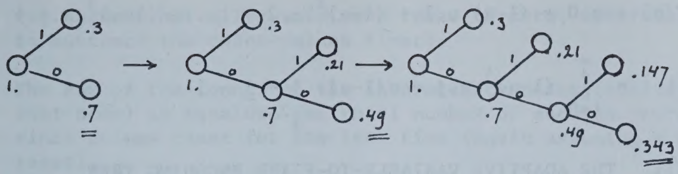
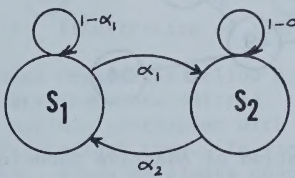


Figure 1: Tunstall's optimal variable-to-fixed algorithm (p = 0.30, n = 4)

We consider a *discrete source having a finite number of states* $S_i \in S$. *In each state*, the source behaves like a *discrete memoryless source*. The source is *first-order Markov*, i.e.,

$$P_r\{S(t)|S(t-1),\dots,S(t-k)\} = P_r\{S(t)|S(t-1)\}, k \geq 1$$

After each source symbol that is emitted, the source may change state, but independently of this emitted source symbol. Therefore, it is not possible to identify a unique sequence of states when the output sequence is observed. Such sources are said to be *non-unifilar*[6,7,8]. A classical example of such a source is *Gilbert's 2-state source*, depicted in a somewhat more general fashion in figure 2. Although recurrence relations for P(m,N) the probability of m 1's occurring in a block of N symbols, have been derived[7,8] for the binary case, no method for finding the source entropy H(U) is known.



(α_1, α_2 transition probabilities)

Figure 2: Variant of Gilbert's Source

Let the matrix $P = \{P_{ij}\}$, where $P_{ij} = P_r\{S(t) = S_i | S(t-1) = S_j\}$ be the state-transition probability matrix, then under the given assumptions, the source is ergodic and we can determine the state probabilities $\Pi_i = P_r\{S_i\}$ from

$$\underline{\Pi} = P\underline{\Pi}$$

$$\sum_{i=1}^{s} \Pi_i = 1$$

where $s = |S|$.

We will in particular be interested in the case $s = 2$ and $P_{12} = P_{21} = \alpha_1 = \alpha_2 = \alpha$. Then $\Pi_1 = \Pi_2 = \frac{1}{2}$. As a measure of how fast the source varies in time, i.e., changes state, we define $T(\alpha)$ as the average time that the source remains in the same state. It follows quite easily that,

$$T(\alpha) = \alpha.0 + (1-\alpha).\alpha.1 + (1-\alpha)^2.\alpha.2 + \ldots + (1-\alpha)^j.\alpha.j + \ldots$$

$$= \sum_{j=0}^{\infty} (1-\alpha)^j.\alpha.j = \alpha(1-\alpha).\,\frac{1}{\alpha^2} = \frac{1}{\alpha} - 1$$

III. THE ADAPTIVE VARIABLE-TO-FIXED ENCODING TREE

Gallagher[5] presented an adaptive Huffman-algorithm in which the decoding-tree (i.e. the codewords) adapted itself to the (slowly) varying source statistics. Since variable-to-fixed codes have better synchronization properties than the fixed-to-variable Huffman codes, we have constructed an adaptive encoding-tree algorithm. To encode a source-sequence the encoding-tree is traversed starting at its root until a terminal node is reached. The branches in the encoding-tree are labeled with the source symbols. With each terminal node is associated a unique codeword. In our adaptive algorithm we keep track of the number of times that a node (root node, intermediate node or terminal node) is traversed, viz. its count. Initially we could set all the counts to zero, but usually we set the counts to a value proportional to the expected node-probabilities. As an example, when we estimate that the source momentarily behaves like a binary memoryless source with $p \simeq 0.3$, while $n = 4$ then we could reset the tree as indicated in figure 3.

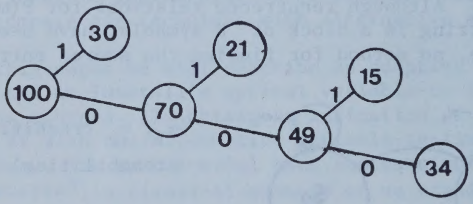


Figure 3: Initialization of Adaptive Encoding Tree

After every encoded message, we start at the root of the tree and depending on the new source symbols we traverse the tree while incrementing the counts of the nodes that we pass. As soon as a terminal node obtains a count as high as the count of some intermediate node, we 'cut' the tree at that intermediate node and change the tree by connecting the sub-tree that extends beyond the intermediate node to the terminal node. The intermediate node becomes now a new terminal node and we associate with it the codeword of the old terminal node.

As an illustration, consider the encoding tree in figure 4a, with messages 1, 01, 001 and 000. Suppose we just encoded a message and the source-sequence continues as 01... . We increment the count of the root node (now 12), take the 0-branch, update the count of corresponding intermediate node (from 9 to 10), continue with the 1-branch to a terminal node, whose count is incremented from 4 to 5. At this point (figure 4b) we see that the intermediate node with lowest count also has a count equal to 5. We now change the encoding-tree, as shown in figure 4~~b~~c and we have a new message-set {1,011,010,00}.

Note that the encoding tree has the following properties:

1) The count of the root node is equal to the total number of codewords sent over the channel since it was reset for the last time

(we assume that all counts were reset to zero, otherwise we have to subtract the reset-values first).

2) The sum of the counts of the intermediate nodes (including the root-node) is equal to the total number of encoded source symbols since it was reset for the last time (again assuming a zero-count reset).

3) The sum of the counts of two sibling nodes is equal to the count of their parent node.

4) The count of a terminal node indicates how often the corresponding codeword has been sent over the channel, since the tree was reset for the last time (again assuming a zero-count reset).

5) The sum of the counts of the terminal nodes equals the count of the root-node.

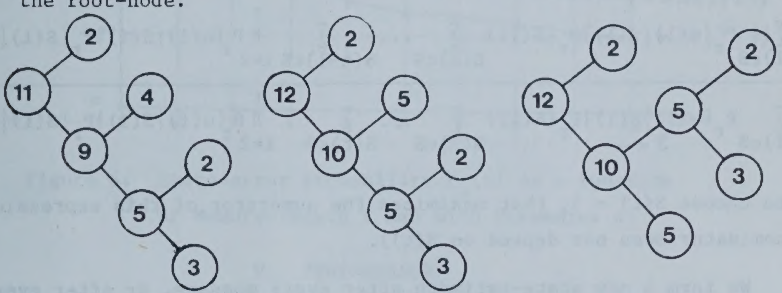


Figure 4: Illustration of ADAP Algorithm

The proofs of these properties follow directly from the described encoding algorithm and are therefore omitted. To prove property 4), note that a terminal node may not correspond with the same codeword it had when the tree was reset for the last time. In fact, we do not necessarily have the same tree. However, since we always change a tree at nodes with equal counts the proof follows.

From properties 1) and 2) we can easily find the rate of the code (since the last reset time) as

$$R = \frac{[\text{count root-node}] * [\text{codeword length } \lceil \log n \rceil\,]}{[\text{sum of the counts of the intermediate nodes + root-node}]}$$

and $\gamma \overset{\Delta}{=} \frac{1}{R}$

is defined as the compression-ratio.

In order to prevent count-overflows and to let the tree adapt fast enough to source-fluctuations we reset the tree at appropriate times.

In the appendix we describe an implementation of the algorithm. Because of its simplicity this algorithm is well suited for implementation on microprocessors.

IV. STATE-ESTIMATION APPROACH

Under the condition that the transition probabilities are small (i.e. slowly varying source) we can form an estimate for the current state of the (non-unifilar) Markov source. We must, however, keep in mind that it is not possible for a non-unifilar source to uniquely identify a state-sequence from the observation of past outputs. We therefore have to 'average' over all 'state-sequences' that can produce the given observed output sequence.

Based on the t most recently encoded source-symbols we form an estimate of the state. For every state S_i we have an (usually different) encoding tree, optimal for that particular state, again constructed according to Tunstall's algorithm with $p = p_i$. We want to find $\hat{S}(t) = S_i$, such that $P_r\{S(t) = S_i | u(t), u(t-1), \ldots, u(1)\}$ is maximized. Using the earlier mentioned properties of the source, we find (for $t \geq 2$) after some algebraic manipulations

$$P_r\{S(t)|u(t),u(t-1),\ldots,u(1)\} =$$

$$\frac{\sum\limits_{S(1)\varepsilon S} P_r\{u(1)|S(1)\}P_r\{S(1)\} \sum\limits_{S(2)\varepsilon S} \cdots \sum\limits_{S(t-1)\varepsilon S} \prod\limits_{i=2}^{t} P_r\{u(i)|S(i)\}P_r\{S(i)|S(i-1)\}}{\sum\limits_{S(1)\varepsilon S} P_r\{u(1)|S(1)\}P_r\{S(1)\} \sum\limits_{S(2)\varepsilon S} \cdots \sum\limits_{S(t)\varepsilon S} \prod\limits_{i=2}^{t} P_r\{u(i)|S(i)\}P_r\{S(i)|S(i-1)\}}$$

Then choose $S(t) = S_j$ that maximizes the numerator of this expression (the denominator does not depend on $S(t)$).

We form a new state-estimate after every message, or after every k^{th} message, but it is important that we never choose for another tree until a message is completed. This way, the decoder is able to find out what encoding-tree is currently in use.

Clearly the probability of an incorrect state-estimation, i.e., $P_e = P_r\{\hat{S}(t) \neq S(t)\}$ plays an important role in how well this algorithm performs. This probability P_e depends on both state-transition probabilities P_{ij} and on the parameter t.

Consider the Gilbert-source of figure 2 and let the source in state S_1 behave as a binary Bernouilli-source with probability $P_r\{u(t)=1|S_1\}=p_1$ and in state S_2 as a binary Bernouilli-source with probability $P_r\{u(t) = 1|S_2\} = p_2$. Furthermore, let $\alpha_1 = \alpha_2 = \alpha$, then $\Pi_1 = \Pi_2 = 0.5$ and $\overline{p} = (p_1 + p_2)/2$

Let $p_1 = .1$ and $p_2 = .5$, then $\overline{p} = 0.3$

After zealously and carefully carrying out the necessary algebra, we find for $P_e = P_r\{\hat{S}(t) \neq S(t)\}$ for $t = 1$ and $t = 2$ respectively

$$P_e(t = 1) = 0.3$$

$$P_e(t = 2) = 0.28 * \alpha + 0.22$$

For $t > 2$ calculations for P_e become very cumbersome, so that we have used a computer-simulation using the random-number generator on the

IBM370/168 to obtain the plot in figure 5, with the parameter α taking on the values 0.01, 0.02 and 0.03, i.e., $T(\alpha)$ takes on the values 99, 49 and 32 respectively. Computer simulations have also indicated that when α becomes larger (say $\alpha > 0.1$, $T(\alpha) < 9$) then making t large ($t > 10$) will not decrease the error probability $P_e(t)$. This is coherent with the fact that the source is non-unifilar. This is the reason why the state-estimation approach only works for relatively slow varying sources.

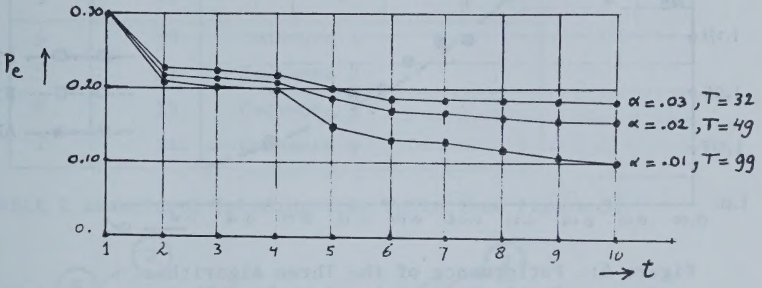


Figure 5: State-error Probability $P_e(t)$ as a Function of Memory-length t and with Parameter α.

V. PERFORMANCE

Extensive computer simulations using the random-number generator of the IBM370/168 have been carried out to test the ADAP algorithm against the state-estimation technique (STAEST) and a fixed-tree algorithm (FIXVFT). The fixed-tree is constructed according to Tunstall's algorithm for an average value $\bar{p}$, given by

$$\bar{p} = \sum_{i=1}^{s} \Pi_i P_i$$

The plot in figure 6 shows the performance of the three algorithms for $n = 16$ and $t = 5$ for STAEST, carried out on source-sequences of 10^5 bits. We have plotted the compression-ratio $\gamma = 1/R$ against α. We see that the state-estimation technique works best for the very slowly varying source ($T \simeq 40$ or larger) while in the 'mid range' ($3 < T < 40$) the adaptive encoding-tree is preferable. Only when the source changes very rapidly ($T \leq 3$) the fixed-tree wins.

VI. CONCLUSIONS

For a specific but frequently used source-model (Gilbert Source) we have tested an adaptive encoding-tree algorithm against two other techniques. Although our results have been mostly qualitative, simulation results indicate that we can consider three ranges: 1) The fast varying source, for which we may as well use a fixed tree, 2) The very slowly varying source, for which we can estimate the state and then use a fixed optimal tree for that state and 3) A 'mid-range' in which the adaptive encoding tree performs best. More analysis needs to be done on both the state-estimation and the adaptive technique.

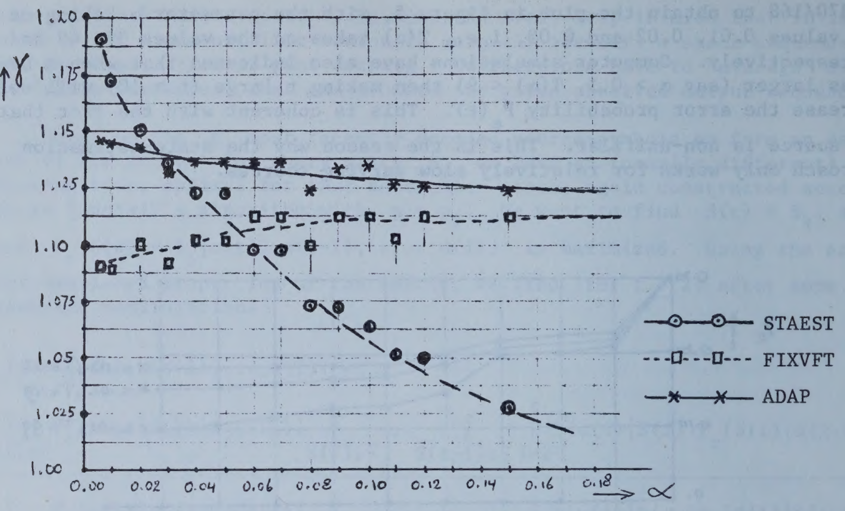


Figure 6: Performance of the Three Algorithms.
Simulation results for the Gilbert-source with $P_1 = 0.1$, $P_2 = 0.5$, n = 16 and t = 5 (for STAEST).

APPENDIX

Implementation and Simulation of ADAP

In this appendix we will describe a possible implementation of the adaptive variable-to-fixed algorithm ADAP as it has been used in the simulations. A few bugs that showed up in the original version were eliminated, the main one being pointed out in this section.

In the case of n codewords, there are n - 1 intermediate nodes (including the root-node) and n terminal nodes. We form a table in which the intermediate nodes, starting with the root node, are the first n - 1 entries, followed by the n terminal nodes.

Associated with each intermediate node is a count CT(i), two tree-pointers FP0(i) and FP1(i) and two list pointers LPU(i) and LPD(i). The forward pointers FP0(i) and FP1(i) are pointing respectively to the 0-sibling and the 1-sibling node of the ith intermediate node. All intermediate nodes are chained in a list. A toplist-pointer points to the intermediate node j with lowest count CT(j). The list pointers LPU(i) and LPD(i) are pointing respectively to the intermediate node with respectively next higher (or equal) and next lower (or equal) count. Each time that we increment the count of an intermediate node we have to check its position in the list and change pointers when necessary. In the case that n is small we can avoid these list pointers and do a linear search after every encoded message to see whether the tree has to change. Associated with every terminal node in the table are only its count CT(i) and a codeword.

In the original algorithm there could occur a problem, viz., just after a tree-change a terminal node could obtain a higher count than the lowest count among the intermediate nodes. As an illustration, consider the encoding tree in figure 7a. The next source-symbol 0 is recognized as a message and the concerning counts are incremented (figure 7b), after which we have to change the tree (figure 7c). The next source-symbol 1 is again recognized as a message and this time incrementing the counts leads to the undesired property. (figure 7d)

Node Number	Count CT	FPO	FP1	LPU	LPD
1	100	2	4	-	2
2	70	3	5	1	3
3	49	7	6	2	-
4	30	Codeword 1			
5	21	Codeword 2			
6	15	Codeword 3			
7	34	Codeword 4			

Toplist-pointer => 3

TABLE I Example of Encoding-tree Table (see figure 3)

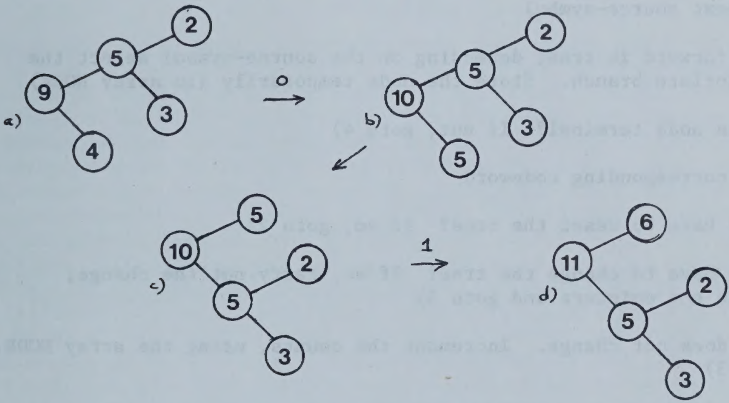


Figure 7: Anomaly in Original ADAP Algorithm

We have avoided this problem by a slight-modification that hardly affects the true tree-statistics for slowly varying sources. Rather than changing the tree when a terminal node obtains the same count as the lowest intermediate-node count, we change when the terminal node would obtain a higher count than the lowest-intermediate-node count (toplist pointer). However, in order to preserve the property that the sum of two sibling nodes equals the count of their parent node we do not carry out the count-incrementing procedure for a message that leads to a tree-change. As an example, suppose we have the encoding tree in figure 8a. Let the next source-symbol be a 1, which is recognized as a message. At this point we must change the tree but we will not increment the counts of the root-node and its 1-sibling node. We then obtain the tree in figure 8b.

In figure 9 the flow diagram of this adaptive encoding tree algorithm (ADAP) is given.

Our discussion has been for a binary source, but it is a straightforward technique to implement the algorithm for non-binary sources. In that case, we obtain pointers FPO, FP1, ..., FPC for a (C+1)-ary source.

We leave further changes to the imagination of the reader.

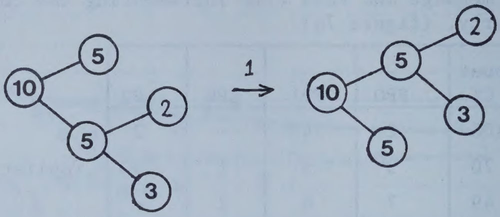


Figure 8: Illustration of Modified ADAP Algorithm

1) Obtain estimated source-statistics (if any available) and read n.

2) Initialize the tree (FP0, FP1, LPU, LPD, Toplist pointer, CT)

3) Start at root-node of tree

4) Get next source-symbol

5) Step forward in tree; depending on the source-symbol select the appropriate branch. Store the node temporarily (in array NODE)

6) Is the node terminal? If not, goto 4)

7) Send corresponding codeword

8) Do we have to reset the tree? If so, goto 2)

9) Do we have to change the tree? If so, carry out the change, update the pointers and goto 3)

10) Tree does not change. Increment the counts, using the array NODE. Goto 3)

Remarks: Before we change the encoding tree, we have to send the codeword in order to enable the decoder to carry out the same change.

Figure 9: Flow-diagram of ADAP Algorithm
Note that step 8) depends on agreements made with the decoder.

REFERENCES

1. B. D. Tunstall, "Synthesis of Noiseless Compression Codes," Ph.D. Thesis, Dept. of Electrical Engineering, Georgia Institute of Technology, 1968.

2. F. Jelinek and K. S. Schneider, "On Variable-Length-to-Block Coding," IEEE Trans. Inform. Theory, vol. IT-18, pp 765-774, Nov. 1972.

3. D. L. Cohn, "Optimum Noiseless Source Codes with Fixed Dictionary Size," presented at the 1976 International Symposium on Information Theory, Ronneby, Sweden, June 1976.

4. R. G. Gallager, Information Theory and Reliable Communication, New York, Wiley, 1968.

5. R. G. Gallager, "Adaptive Huffman Codes," presented at the 1977 International Symposium on Information Theory, Ithaca, New York, October 1977.

6. J. L. Massey, "Markov Information Sources," presented at the NATO Advanced Study Institute on New Directions in Signal Processing in Communications and Control, Darlington, England, August 5-17, 1974.

7. L. N. Kanal and A. R. K. Sastry, "Models for Channels with Memory and Their Applications to Error Control," Proc. IEEE, vol. 66, No. 7, pp. 724-744, July 1978.

8. E. N. Gilbert, "Capacity of a Burst-Noise Channel," Bell Syst. Tech. J., vol. 39, pp. 1253-1266, September 1960.

QUANTIZATION OF BIVARIATE CIRCULARLY SYMMETRIC DENSITIES

J. A. BUCKLEW & N. C. GALLAGHER
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

The problem of quantizing a two dimensional random variable whose bivariate density has circular symmetry is considered in detail. Two quantization methods are considered, leading to polar and rectangular representations. A simple necessary and sufficient condition is derived to determine which of these two quantization schemes is best. If polar quantization is deemed best, the question arises as to the ratio of the number of phase quantizer levels to that of magnitude quantizer levels when the product of these numbers is fixed. A simple expression is derived for this ratio that depends only upon the magnitude distribution. Several examples of common circularly symmetric bivariate densities are worked out in detail using these expressions.

I. INTRODUCTION

Consider a two dimensional random variable X whose bivariate density is circularly symmetric and we desire to represent this quantity by a finite set of values. One possible representation of X leads to a Cartesian co-ordinate system expression wherein we individually quantize the two rectangular components of the random variable. Another common representation leads to a polar co-ordinate representation where we quantize the magnitude and phase angle of X. These two representations are mainly chosen for their computational feasibility and ease of implementation. Other authors have considered the general problem of multidimensional quantization; Zador [1] derives an expression for the minimum error achievable by a multidimensional quantizer for an arbitrary density, but no insight into the required quantizer structure is attained. Chen [2] describes a technique whereby one can use a recursive computer technique to solve for a "good" quantizer, but the optimality of the final solution is not assured. By constraining ourselves to circularly symmetric densities and also to either Cartesian or polar co-ordinate systems, it becomes possible to reduce the optimal two dimensional quantization problem to one dimension. Max [3] develops necessary conditions for the optimality of a one dimensional quantizer. Panter and Dite [4], give a formula for the asymptotic error to be expected for optimal mean square error quantizers (of sufficiently smooth input densities).

In Section II of this paper we obtain a simple criterion by which to determine whether polar format or rectangular format gives a smaller mean square quantization error. It is shown for some very important cases, notably for the Gaussian bivariate density, that polar format is asymptotically superior.

If polar format is to be used and the product $N = N_\theta N_r$ is fixed, where N_θ and N_r are the number of phase and magnitude quantization levels, respectively, the question arises as to the optimum ratio N_θ / N_r. We derive a simple expression for this ratio that depends only upon the magnitude density.

In Section III, we provide several examples of common circularly symmetric densities (e.g. marginal densities are Pearson II, Pearson VII, sinusoidal, and Gaussian) and we address the question of whether the rectangular or the polar format scheme gives a smaller quantization error.

II. DEVELOPMENT

Consider the mean square quantization error E_p of a polar format representation,

$$E_p = \sum_{j=1}^{N_\theta} \sum_{i=1}^{N_r} \int_{c_{j-1}}^{c_j} \int_{a_{i-1}}^{a_i} |re^{j\theta}-b_i e^{jd_j}|^2 \, \frac{f_r(r)\, dr\, d\theta}{2\pi} \,. \tag{1}$$

Implicit use has been made of the fact that in circularly symmetric bivariate densities the magnitude random variable with probability density $f_r(\cdot)$ is independent of the uniformly distributed $[-\pi,\pi]$ phase random variable. The b_i and d_j are the output levels of the magnitude and phase quantizers corresponding to input levels lying in the intervals $(a_{i-1}, a_i]$ and $(c_{j-1}, c_j]$, respectively. It is shown in [5] that the optimal phase quantizer is the uniform quantizer. This allows us to simplify Eq. (1);

$$E_p = \sum_{i=1}^{N_r} \int_{a_{i-1}}^{a_i} [r^2+b_i^2-2rb_i \frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}}] \, f(r)\, dr \,. \tag{2}$$

Differentiating with respect to b_i, we find the optimum b_i is

$$b_i = \frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}} \; \frac{\int_{a_{i-1}}^{a_i} rf(r)\, dr}{\int_{a_{i-1}}^{a_i} f(r)\, dr} \,. \tag{3}$$

The equation given by Max for the output levels b_i' of an optimal one dimensional magnitude quantizer is found in [3] to be

$$b_i' = \frac{\int_{a_{i-1}'}^{a_i'} rf(r)\, dr}{\int_{a_{i-1}'}^{a_i'} f(r)\, dr} \,, \tag{4}$$

where the optimal input interval endpoints a_i' (for the one dimensional case) satisfy

$$a_i' = \frac{b_i' + b_{i+1}'}{2} \,. \tag{5}$$

If we minimize Eq. (2) with respect to the a_i, we then arrive at the necessary condition (for the two dimensional case)

$$a_i = \frac{b_i + b_{i+1}}{2\left(\frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}}\right)} = \frac{b_i' + b_{i+1}'}{2} = a_i' \tag{6}$$

This equation indicates that the quantizer interval endpoints for the optimum magnitude quantizer in the two dimensional case is the same as the quantizer interval endpoints for the optimum one dimensional quantizer. From Eqs. (3) and (4) and the preceeding discussion, we have the following relationship between the output levels b_i' and b_i:

$$b_i' = \frac{\frac{\pi}{N_\theta}}{\sin\frac{\pi}{N_\theta}}\, b_i \ . \tag{7}$$

Consequently, Eq. (2) becomes

$$E_p = E\{r^2\} - \left(\frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}}\right)^2 \sum_{i=1}^{N_r} (b_i')^2 \int_{a_{i-1}}^{a_i} f(r)\, dr \ , \tag{8}$$

where $E\{\cdot\}$ is the statistical expectation operator. In [6] it is shown that the mean square quantization error for a minimum mean square error quantizer is simply the input variance minus the output variance. If we denote by E_X^N the mean square quantization error produced by an optimal N level quantizer for the random variable X, we may rewrite Eq. (8) as

$$E_p = \left(\frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}}\right)^2 E_r^{N_r} + \left(1 - \left(\frac{\sin\frac{\pi}{N_\theta}}{\frac{\pi}{N_\theta}}\right)^2\right) E\{r^2\} \ . \tag{9}$$

Our problem now is one of characterizing the quantity E_X^N. Panter and Dite [4] give a formula for the expected error of a minimum mean square error quantizer with a large number of output levels and a smooth input density. This formula is

$$E_X^N = \frac{K_X}{N^2}$$

where

$$K_X = \frac{\left[\int_{-\infty}^{\infty} f(x)^{1/3}\, dx\right]^3}{12} \ . \tag{10}$$

Roe [7] also derives some asymptotic formula which were later used by Wood [8] to rederive Eq. (10). Roe's formula depend on the truncation of a Taylor series expansion of the input density. Wood, in his formula, explicitly states that the input density and the first few derivates (up to order five in some cases) must exist and be continuous. Panter and Dite, in their derivation, require that as the input intervals become very small, the density function may be approximated as a constant over each interval. In [1] it is shown that a sufficient condition for Eq. (10) to hold is that $f(x)$ be Riemann integrable and that $E\{x^{2+\delta}\} < \infty$ for some $\delta > 0$, in general a much less severe restriction than continuity or differentiability.

We make use of the approximation

$$\left(\frac{\sin x}{x}\right)^2 \cong 1 - \frac{x^2}{3} \ , \tag{11}$$

and of Eq. (10) in order to reduce Eq. (9) as

$$E_p \cong \left(1 - \frac{\pi^2}{3N_\theta^2}\right) \frac{K_r}{N_r^2} + \frac{2}{3}\frac{\pi^2}{N_\theta^2} \; . \tag{12}$$

If we denote as N the total number of output levels allowed to represent the two dimensional random variable X, we have the following relation,

$$N = N_r \, N_\theta \; . \tag{13}$$

Since $K_r > 0$, it is simple to show that $N_r = 0(N^{1/2})$ and $N_\theta = 0(N^{1/2})$ by differentiating Eq. (12) and solving for the optimal quantities. Making use of this fact and Eq. (13), we may, assuming sufficiently large N, write Eq. (10) as

$$E_p = \frac{K_r N_\theta^2}{N_\theta^2} + \frac{2}{3}\frac{\pi^2}{N_\theta^2} \; . \tag{14}$$

This is then optimized with respect to N_θ and yields the optimal N_θ^2 as

$$N_\theta^2 = \left(\frac{2\pi^2}{3K_r}\right)^{1/2} N \; . \tag{15}$$

This leads to the following expression for the minimal attainable asymptotic polar format error,

$$(E_p)_{opt} = \sqrt{\frac{2K_r}{3}\frac{2\pi}{N}} \; . \tag{16}$$

Now consider the problem of optimally quantizing the random variable X in a rectangular format. The mean square quantization error, E_x, of this representation is given by

$$E_x = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \int_{g_{j-1}}^{g_j} \int_{e_{i-1}}^{e_i} [(x-f_i)^2 + (y-h_j)^2] \, f_{x,y}(x,y) \, dxdy \; , \tag{17}$$

where N_x and N_y are the number of levels in each of the respective orthogonal random variables. The other notation should be clear. Equation (17) may be written as

$$E_x = \sum_{i=1}^{N_x} \int_{e_{i-1}}^{e_i} (x-f_i)^2 \, f_x(x) \, dx + \sum_{j=1}^{N_y} \int_{g_{j-1}}^{g_j} (y-h_j)^2 \, f_y(y) \, dy \; . \tag{18}$$

By symmetry arguments (since $f_x(x) = f_y(x)$), we may argue that $N_x = N_y = N^{1/2}$. The quantizer that minimizes the above equation is simply the minimum mean square error quantizer for each of the two components. Therefore, again using Eq. (10) we have for large N.

$$E_x = \frac{2K_x}{N} \; ,$$

where

$$K_x = \frac{\left[\int_{-\infty}^{\infty} f(x)^{1/3} \, dx\right]^3}{12} \; . \tag{19}$$

Comparing Eq. (19) and Eq. (16), we say that polar format is asymptotically better than rectangular format if and only if

$$\frac{2K_x}{N} > \sqrt{\frac{2K_r}{3}}\,\frac{2\pi}{N} ,$$

or

$$K_x > \sqrt{\frac{2K_r}{3}}\,\pi \quad . \tag{20}$$

In other words, if the inequality is satisfied and the original input probability density is Riemann integrable, then we are guaranteed that there exists an N_0 such that for every $N > N_0$, polar format quantization will perform better than rectangular format quantization.

If polar quantization is deemed best for a particular density, then what is the ratio N_θ/N_r that provides the smallest total error? This question is answered with the use of Eq. (15); we find

$$\left(\frac{N_\theta^2}{N}\right)_{opt} = \left(\frac{N_\theta}{N_r}\right)_{opt} = \sqrt{\frac{2}{3K_r}}\,\pi \ . \tag{21}$$

III. EXAMPLES

For our first example, we calculate the relevant parameters for a random variable whose marginal density is of Pearson Type VII. This distribution is a generalization of Students-t distribution. The bivariate density is

$$f(x,y) = \frac{v}{\pi}\,\frac{2^v(v-1)^v}{(2(v-1) + x^2 + y^2)^{v+1}} , \quad -\infty < x,\ y < \infty \tag{22}$$

(with $v > 1$ in order to assure finite variance) and the marginal density appears as

$$f(x) = \frac{2^v(v-1)^v\ \Gamma(v+1/2)}{\sqrt{\pi}\ \Gamma(v)\ (2(v-1) + x^2)^{v+1/2}} , \quad -\infty < x < \infty \tag{23}$$

where $\Gamma(\cdot)$ is the gamma function and where we have normalized the distribution so that $f(x)$ has unit variance. The magnitude density is always derived by substituting in r for $\sqrt{x^2 + y^2}$ in $f(x,y)$ and multiplying the result by $2\pi r$, as shown by a simple change of variable. Eq. (23) yields after some tedious algebra

$$K_x = \frac{[B(\frac{1}{2}\ ;\ \frac{v-1}{3})]^3}{12\ B(\frac{3}{2}\ ;\ v-1)} , \tag{24}$$

where $B(\cdot;\ \cdot)$ is the beta function. We perform similar operations with the magnitude density to yield

$$K_r = \frac{v(v-1)}{24}\,[B(\tfrac{2}{3}\ ;\ \tfrac{v-1}{3})]^3 \ . \tag{25}$$

In Fig. 1 K_x (solid line) and $\sqrt{\frac{2K_r}{3}}\,\pi$ (dotted line) are plotted as a function of v for values from 1.1 to 21.1. As shown by this graph, polar format is always asymptotically best for this class of distributions. An interesting point about this set of distributions is that in the limit as $v \to \infty$ Eq. (23) converges to a unit variance Gaussian density. Therefore, taking this limit in Eq. (24) and making use of Stirling's approximation, we have

$$K_x \to \frac{\sqrt{3\pi}}{2} \cong 2.721 \quad . \tag{26}$$

Wood [8] estimates this number as 2.73 which is close to our derived value. From Eq. (25) we have similarly

$$K_r \to \frac{3}{8}\,(\Gamma(\tfrac{2}{3}))^3 \cong .931 \quad , \tag{27}$$

which is the parameter for the Rayleigh distribution obtained in the limit. Using these two values in Eq. (20), we conclude that asymptotically polar formatting is better than rectangular formatting for Gaussian bivariate densities. As a matter of interest, when we substitute the value of K_r found in Eq. (27) into Eq. (21), we find the optimal ratio N_θ/N_r to be 2.659. Pearlman [9] using distortion rate theory states that this ratio should be > 2.596, which is in agreement with our result.

For the next example, consider distributions of the Pearson II class. The bivariate density is

$$f(x,y) = \frac{v(2(v+1) - (x^2+y^2))^{v-1}}{\pi 2^v (v+1)^v}\, U(2(v+1) - (x^2+y^2)) \quad , \tag{28}$$

where $v > 0$, and $U(\cdot)$ is the unit step function. The marginal density is

$$f(x) = \frac{\Gamma(v+1)(2(v+1) - x^2)^{v-\frac{1}{2}}\, U(2(v+1) - x^2)}{2^v (v+1)^v \sqrt{\pi}\, \Gamma(v+\frac{1}{2})} \quad . \tag{29}$$

For $v = 1/2$ we find that $f(x)$ has a uniform distribution. For $v = 1$, we have that the bivariate density is uniform over a circular region in the plane. Using Eq. (29), we find

$$K_x = \frac{[B(\frac{1}{2}\ ;\ \frac{2v+5}{6})]^3}{12\, B(\frac{3}{2}\ ;\ v+\frac{1}{2})} \quad . \tag{30}$$

From the magnitude density we derive that

$$K_r = \frac{v(v+1)}{24}\, [B(\tfrac{2}{3}\ ;\ \tfrac{v+2}{3})]^3 \quad . \tag{31}$$

In Fig. (2) can be seen a plot of K_x (solid line) and $\sqrt{\frac{2}{3}}\, K_r\, \pi$ (dotted line) as a function of v for values from 0 to 10. It should be noted that Eq. (29) also converges to a Gaussian density as $v \to \infty$. It is a simple matter to check that the expressions in Eqs. (30) and (31) indeed go to the correct limits. From the plot it can be seen that for values of v in the interval (0.0, .4) polar format is better. In the interval (.4, 3.635) it is seen that rectangular is better, and from 3.635 to ∞ polar again is better. It appears then that for the circularly symmetric bivariate density whose marginal density is uniform, we have the interesting result that rectangular format is asymptotically better than polar format.

In the theoretical development and in the examples considered so far, we have constrained the class of quantizers considered to two different types, the rectangular format and the polar format. In general, neither of these schemes will be optimal for an arbitrary two dimensional random variable with a circularly symmetric probability density. Zador [1] gives an expression for the asymptotic mean square error E_Z of the optimal two dimensional mean square error quantizer. This equation is

$$E_z = C_z/N \ , \tag{32}$$

where

$$C_z = \frac{5}{18\sqrt{3}} \left[\int_{\infty}^{\infty} \int f_{x,y}(x,y)^{1/2} \, dxdy \right]^2 \tag{33}$$

For the Pearson VII density $C_z = 4.0307\ v/(v-1)$, for the Pearson II density $C_z = 4.0307\ v/(v+1)$. Since in the limit as v becomes large, both of these classes of densities converge to the Gaussian, the smallest error attainable for a two dimensional normal random variable is approximately 4.0307/N. The best that we can do with a polar format representation is 4.95/N and the best that we can do with a Cartesian format representation is 5.442/N. There is certainly room for improvement here, however, the important thing to note is that the structure of the polar format quantizer is known while that of the theoretical optimum quantizer is not.

In section II it was stated that a sufficient condition for Eq. (10) to be valid is that the magnitude density function be Riemann integrable. For most density functions of interest in modeling physical systems this criterion is met. One group of densities that doesn't meet this condition is the set of atomic densities, i.e., densities for which probability mass is contained at a single point. In a circularly symmetric bivariate density, the phase must be uniformly distributed $[-\pi,\pi]$. The only quantity that can be discrete is the magnitude distribution, i.e. we may have "rings" of probability mass distributed in the plane. Suppose we have a single "ring" of probability mass, where the radius of the ring is 1, i.e.,

$$F(r) = U(r-1) \quad , \tag{34}$$

where $F(\cdot)$ is the magnitude distribution function and $U(\cdot)$ is the unit step function. The rectangular component marginal density is the sinusoidal density

$$f(x) = \frac{U(1-x^2)}{\pi\sqrt{1-x^2}} \quad . \tag{35}$$

This density function is Riemann integrable, hence Eq. (10) and Eq. (19) are valid. This implies the rectangular format error is $O(N^{-1})$. Now consider the polar format case. For $N_r \geq 1$, $E_r^{N_r} = 0$. This implies the polar format error for large N is $O(N^{-2})$. Clearly then polar format is asymptotically better for this density. By extending this argument, we may say that if $P(r=0) \neq 1$, then for any bivariate circularly symmetric density with an atomic magnitude density with a finite number of atoms, polar format will give a smaller asymptotic mean square quantization error than rectangular format.

IV. SUMMARY

In this paper we have derived a simple criterion to determine whether rectangular format or polar format gives smaller mean square error for circularly symmetric densities. The optimal ratio of phase quantizer levels to magnitude quantizer levels is also derived. Several examples including the Gaussian case have been studied in detail.

It is interesting to note that polar format is not always better than rectangular format even for the case of densities with circular symmetry.

This research has been supported by the Air Force Office of Scientific Research under grant AFOSR 78-3605.

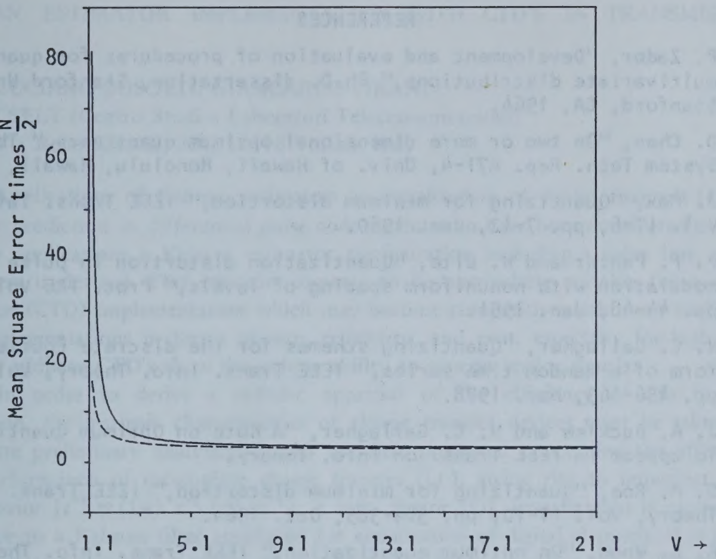


Fig. 1. The solid line is a plot of K_x as a function of V, the dotted line is a plot of $\sqrt{\frac{2K_r}{3}}\,\pi$ as a function of V for the Pearson VII density.

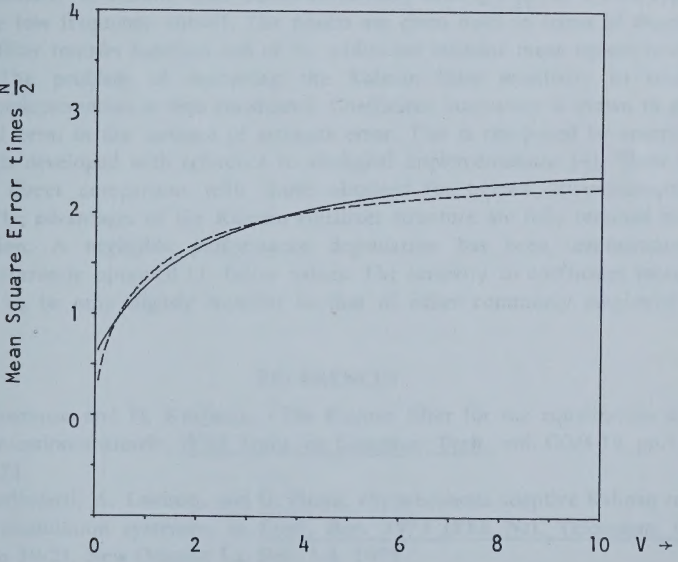


Fig. 2. The solid line is a plot of K_x as a function of V, the dotted line is a plot of $\sqrt{\frac{2K_r}{3}}\,\pi$ as a function of V for the Pearson II density.

REFERENCES

[1] P. Zador, "Development and evaluation of procedures for quantizing multivariate distributions," Ph.D. dissertation, Stanford University, Stanford, CA, 1964.

[2] D. Chen, "On two or more dimensional optimum quantizers," The Aloha System Tech. Rep. A71-4, Univ. of Hawaii, Honolulu, Hawaii, Jan. 1971.

[3] J. Max, "Quantizing for minimum distortion," IEEE Trans. Info. Theory, Vol. IT-6, pp. 7-12, Jan. 1960.

[4] P. F. Panter and W. Dite, "Quantization distortion in pulse code modulation with nonuniform spacing of levels," Proc. IRE Vol. 39, pp. 44-48, Jan. 1951.

[5] N. C. Gallagher, "Quantizing schemes for the discrete Fourier transform of a random time series," IEEE Trans. Info. Theory, Vol. IT-24, pp. 156-163, Mar. 1978.

[6] J. A. Bucklew and N. C. Gallagher, "A Note on Optimum Quantization," To appear in IEEE Trans. on Info. Theory.

[7] G. M. Roe, "Quantizing for minimum distortion," IEEE Trans. Info. Theory, Vol. IT-10, pp. 384-385, Oct. 1964.

[8] R. C. Wood, "On optimum quantization," IEEE Trans. Info. Theory, Vol. IT-5, pp. 248-252, Mar. 1969.

[9] W. A. Pearlman, "Quantization Error Bounds for Computer Generated Holograms," Tech. Rep. #65031-1, Stanford University Information Systems Laboratory, Stanford, CA, August 1974.

ON KALMAN ESTIMATOR IMPLEMENTATION WITH CTD'S IN TRANSMISSION SYSTEMS

LUCIANO BOSOTTI, GIANCARLO PIRANI
CSELT (Centro Studi e Laboratori Telecomunicazioni)
Via Reiss Romoli, 274, 10148 Torino, Italy

Applications of Kalman estimators to equalization of digital channels [1], [2] and to linear prediction in differential pulse code modulation have been considered recently [3]. In these applications a Kalman estimator configuration including a delay line, adders and multipliers is derived. This structure appears to be particularly suitable for a charge transfer device (CTD) implementation which may become competitive with more traditional hardware implementations in terms of size, reliability, and cost, especially for high order structures. In addition CTD's show the unique ability to process analog signals.

In order to derive a realistic appraisal of the efficiency of this form of implementation, the intrinsic characteristics of charge transfer devices must be taken into account. Some preliminary analytical results are first presented concerning the effects on estimator performance of incomplete charge transfer (i.t.), giving rise to imperfect delay element behavior [$z^{-1} \Rightarrow (1-\epsilon)\ z^{-1} / (1-\epsilon z^{-1})$; ϵ = i.t. factor (1)]. Investigation is carried out with reference to a Kalman filter employed for equalization of digital channels. It is shown that, in the general case, the mapping expressed by (1) does not affect the stability of the filter. The influence of i.t. on the Kalman equalizer transfer function is shown to be practically negligible for small values of the incomplete transfer factor ($\epsilon < 10^{-3}$); which are compatible with present-day technology.

Computations have been carried out with reference to two channel models for digital transmission: a) a sixth order Butterworth filter with $f_c = \frac{1}{3T}$, b) a PCM type channel with a severe low frequency cut-off. The results are given both in terms of degradation of the Kalman filter transfer function and of the additional estimate mean square error.

The problem of evaluating the Kalman filter sensitivity to tolerances in coefficient implementation is then considered. Coefficient inaccuracy is shown to give rise to an additional term in the variance of estimate error. This is computed by resorting to the same methods developed with reference to all-digital implementations [4]. These results are reported in direct comparison with those obtained for tapped delay-line-equalizers. In conclusion, the advantages of the Kalman equalizer structure are fully retained by the CTD implementation. A negligible performance degradation has been demonstrated in the presence of currently obtained i.t. factor values. The sensivity to coefficient inaccuracy has been found to be only slightly superior to that of other commonly employed equalizer structures.

REFERENCES

[1] R.E. Lawrence and H. Kaufman, «The Kalman filter for the equalization of a digital communication channel», IEEE Trans. on Commun. Tech., vol. COM-19, pp.1137-1141, Dec. 1971.

[2] A. Guardincerri, A. Luvison, and G. Pirani, «Synchronous adaptive Kalman receiver for carrier modulation systems», in Conf. Rec. 1975 IEEE Nat. Telecomm. Conf., pp. 39-19 to 39-23, New Orleans, La, Dec. 1-3, 1975.

[3] J.E. Gunn and A.P. Sage, «Speech data rate reduction. Part I: applicability of modern estimation theory», IEEE Trans. on Aer. and Elect. Syst., vol. AES-9, pp. 130-138, March 1973.

[4] J.B. Knowles and E.M. Olcayto, «Coefficient accuracy and digital filter response», IEEE Trans. on Circuit Theory, vol. CT-15, pp. 31-41, March 1968.

INDEX OF AUTHORS

Abraham, J. 155, 156
Abram, J. M. 271
Adedotun, E. A. 941
Aguilar-Martin, J. 933
Ahmed, M. 30
Aho, A. V. 54
Ali, A. M. 382
Alvarado, F. L. 852
Anderson, J. B. 261, 725, 961
Athay, T. M. 882
Azim, S. A. 202
Baker, B. S. 626
Barmish, B. R. 529
Belanger, P. R. 30, 913
Bentley, J. L. 40, 843
Bhullar, J. P. 261
Bistritz, Y. 903
Bitner, J. R. 224
Blanke, M. 364
Bonnell, R. D. 793
Book, R. V. 489
Booth, K. S. 607
Bosotti, L. 991
Bredeson, J. G. 146
Breitbart, Y. 636
Briggs, F. A. 14
Bruno, J. 232
Buchner, M. R. 1
Bucklew, J. A. 928
Buric, M. R. 655
Cheston, G. A. 278
Chikte, S. D. 89
Chow, J. H. 679
Chyung, D. H. 792
Coffman, E. G., Jr. 626
Cohn, D. L. 972
Coraor, L. D. 585
Cowley, R. A. 185
Coy, W. 314
Crevier, D. 306
Daut, D. G. 392
Davisson, L. D. 697
DeClaris, N. 185
DeSantis, R. M. 914, 933
Desoer, C. A. 521
Doyle, J. C. 461
Drysdale, R. L., III 833
Dunham, J. G. 696
Dussault, J. A. 147
Elrazaz, Z. 762
El-Sheikh, T. S. 278
El-Sherief, H. 923
Eyman, E. D. 941
Finamore, W. A. 712
Fischl, R. 894
Fix, S. L. 704
Fogel, E. 802
Franklin, J. B. 862
Friedman, D. P. 578
Fu, K. S. 165
Gabow, H. N. 242
Gallagher, N. C. 117, 127, 982
Gonzalez, T. F. 64
Groom, N. J. 780
Hadidi, M. T. 79
Hepburn, J. S. A. 547
Heydt, G. T. 862
Hirschberg, D. S. 50
Houpt, P. K. 20
Huang, T. C. 413
Huhns, M. H. 793
Jain, P. C. 413
Johnson, D. B. 64
Johnson, T. L. 20

Jones, J. 232
Joshi, S. M. 780
Kahne, S. 334
Kanellakis, P. C. 214
Kintala, C. M. R. 480
Klein, G. 770
Koivo, A. J. 14
Koivo, H. 744
Koralek, R. W. 354
Kosaraju, S. R. 495
Kwatny, H. G. 882
Kwon, W. H. 670
Lamont, G. B. 287
Langeland-Knudsen, J. 364
Langholz, G. 903
Larimer, S. J. 287
Laub, A. J. 432
Lee, D. T. 833
Lee, P. K. 345
Lefebvre, S. 914, 933
Leon-Garcia, A. 697
Lewis, H. R. 513
Liberty, S. R. 754
Liepa, P. E. 157
Lipton, R. J. 607
Liu, B. 372
Liu, H. L. 165
Lo, J. T.-H. 89
LoCicero, J. L. 951
Loh, N. K. 792
Luccio, F. 73
Lugannani, R. 403
Malik, N. R. 391
Manacher, G. K. 824
Mayewski, J. J. 324
Maynard, R. A. 754
McCannon, T. E. 117, 127
McElhone, T. P. 364
Meditch, J. S. 192
Meisel, J. 870
Metze, G. 155, 156
Metzler, M. A. 354
Michel, A. N. 557
Mielke, R. R. 754
Mikhail, S. H. 664
Miller, R. K. 557
Milstein, L. B. 423
Minoo-Hamedani, D. 117, 127
Modestino, J. W. 392
Mohan, S. 725, 961
Montgomery, R. C. 213
Moore, B. C. 443, 770
Moraga, C. 316
Mudge, T. N. 596
Mugnai, C. 73
Mulder, L. Th. N. M. 972
Munson, D. C., Jr. 372
Murata, T. 175
Nair, R. 155, 156
Nakamae, E. 325
Neuhoff, D. L. 735
Nomura, T. 325
Nourmoussavi, M. A. 306
O'Connor, D. A. 539
Omura, J. K. 413
Papadimitriou, C. H. 214
Payne, A. N. 296, 521
Pearlman, W. A. 712
Pearson, A. E. 670
Pirani, G. 991
Pohjolainen, S. 744
Polak, E. 296
Prezas, D. P. 951
Probert, R. L. 597
Pudlak, P. P. 504
Reddy, S. M. 137
Renjen, S. 793
Rhodes, I. B. 271, 344

Rivest, R. L. 626
Robinson, J. P. 315, 585
Rosenthal, A. S. 136, 252, 616
Safonov, M. G. 451
Sagiv, Y. 54
Sain, M. K. 567
Salut, G. 933
Sandell, N. R., Jr. 471
Schafer, R. M. 567
Schmitendorf, W. E. 529
Schwartz, S. C. 79
Shaw, L. 689
Siegel, H. J. 586
Siegel, L. J. 971
Sinha, A. S. C. 814
Sinha, N. K. 202, 762, 923
Sira-Ramirez, H. 566
Smith, J. E. 138
Smith, R. G. 882
Snyder, D. L. 344
So, K. 232
Springsteel, F. N. 504
Srinathkumar, S. 754
Stein, G. 461
Suk, D. S. 137
Sutton, J. C., III 146
Swamy, K. N. 645
Szymanski, T. G. 54
Tabak, D. 213
Tarn, T. J. 539, 645
Ullman, J. D. 54
Vairavan, K. 568
Vairavan, V. 568
Wahid, J. 20
Wang, Y. T. 521
Ward, S. A. 20
Weide, B. W. 843
Wise, D. S. 578
Wise, G. L. 117, 127
Wong, C. K. 224
Wonham, W. M. 157, 547, 664
Wotschke, D. 480
Wozny, M. J. 11
Wrathall, C. 489
Yamashita, H. 325
Yan, T. Y. 99
Yao, A. C. 843
Yao, K. 99, 423
Yokomoto, C. F. 814
Yu, S. H. 175
Yue, O.-C. 107
Zaborszky, J. 880
Zobrist, A. L. 824

BOUND BY
HNM

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C001
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1978
3 0112 008149137